



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Sistema d'Informació per la Gestió dels Estudiants de Mobilitat Internacional

Autora: Florencia Rímolo Figueira

Director: Jaume Moral Ros

Ponent: Maria Ribera Sancho Samso

Titulació: Grau en Enginyeria Informàtica

Especialitat: Enginyeria del Software

Data de defensa: 21/10/2019 – 15:00h

Modalitat B. Treball realitzat amb:



Resum

Aquest document correspon a la memòria del Treball Final de Grau desenvolupat amb un conveni de cooperació educativa amb l'empresa inLab de la Facultat d'Informàtica de Barcelona.

El projecte consisteix en un sistema d'informació que gestiona les sol·licituds dels estudiants que desitgen realitzar una estància acadèmica a la FIB, des que la universitat origen ens comunica les seves dades personals fins que són admesos per la nostra facultat. Amb aquest sistema es pretén facilitar les tasques a l'Àrea de Relacions Internacionals de la FIB, qui principalment es beneficiarà del producte final. Actualment, aquesta unitat treballa amb una aplicació web que no cobreix al 100% les seves necessitats i implementa una tecnologia que s'ha quedat obsoleta per la qual cosa dificulta el desenvolupament de l'aplicació i, alhora, la fa poc mantenible. A més, el procés actual genera un volum de documentació en paper innecessari avui en dia que implica un impacte ambiental negatiu que, encara que sigui relatiu, es pot estalviar.

Estudiarem també la possible incorporació en un futur d'alguna de les funcionalitats que ofereix la xarxa *Erasmus Without Paper*, un projecte europeu que té com a objectiu eliminar la documentació en paper en l'àmbit de la mobilitat acadèmica oferint diverses APIs amb les quals les universitats es poden intercanviar dades.

Resumen

Este documento corresponde a la memoria del Trabajo Final de Grado desarrollado con un convenio de cooperación educativa con la empresa inLab de la Facultad de Informática de Barcelona.

El proyecto consiste en un sistema de información que gestiona las solicitudes de los estudiantes que desean realizar una estancia académica en la FIB, desde que la universidad origen nos comunica sus datos personales hasta que son admitidos por nuestra facultad. Con este sistema se pretende facilitar las tareas en el Área de Relaciones Internacionales de la FIB, quien principalmente se beneficiará del producto final. Actualmente, esta unidad trabaja con una aplicación web que no cubre al 100% sus necesidades e implementa una tecnología que se ha quedado obsoleta por lo que dificulta el desarrollo de la aplicación y, a la vez, la hace poco mantenible. Además, el proceso actual genera un volumen de documentación en papel innecesario hoy en día que implica un impacto ambiental negativo que, aunque sea relativo, se puede ahorrar.

Estudiaremos también la posible incorporación en un futuro de alguna de las funcionalidades que ofrece la red *Erasmus Without Paper*, un proyecto europeo que tiene como objetivo eliminar la documentación en papel en el ámbito de la movilidad académica ofreciendo varias APIs con las cuales las universidades se pueden intercambiar datos.

Abstract

This document corresponds to the report of the Final Project of Degree developed with an agreement of educational cooperation with the company inLab of the Barcelona School of Informatics.

The project consists of an information system that manages the applications of the students that wish to make an academic stay at the FIB, since the home university communicates their personal data until they are admitted by our faculty. This system aims to facilitate tasks in the International Relations Department of the FIB, who will mainly benefit from the final product. Currently, this unit works with a web application that does not cover its needs 100% and implements a technology that has become obsolete, which makes it difficult for the application to develop and, at the same time, makes it unmanageable. In addition, the current process generates a volume of unnecessary paper documentation, which implies a negative environmental impact that, although relative, it can be saved.

We will also study the possible future incorporation of any of the functionalities offered by the Erasmus Without Paper network, a European project that aims to eliminate paper documentation in the field of academic mobility by offering several APIs with which the universities can exchange data.

Agraïments

Finalitza una etapa molt important a la meva vida. Aquest Treball Final de Grau no només és el fruit de 8 mesos dedicats exclusivament a la seva realització, sinó que també és el fruit d'anys d'esforç i aprenentatge de tots els coneixements adquirits durant aquest temps. És per aquest motiu que no voldria tancar aquest període sense agrair a totes aquelles persones que han estat importants al llarg d'aquesta carrera universitària:

A en Jaume Moral, per haver confiat en mi quan ni tan sols jo mateixa hi confiava, per transmetre'm els seus coneixements quan ho he necessitat. No tinc suficients paraules per agrair-te tot el que m'has ajudat a créixer professionalment.

A la Maria Ribera Sancho, per haver acceptat ser la ponent d'aquest projecte i ajudar a que la memòria tingui l'estructura adequada. Gràcies per transmetre'm els coneixements necessaris sobre l'enginyeria del *software*.

Als meus companys de la universitat, en especial a la Jessica Sobreviela, que de companya de classes i de feina s'ha convertit en una amiga. Gràcies per ajudar-me al llarg de tota la carrera, per fer que no decaigui en els moments més durs i per tots els moments viscuts en aquesta facultat.

Per últim, a la meva família, el pilar fonamental sense el qual no hagués pogut dedicar aquests anys a aquesta disciplina que tant m'apassiona. Gràcies pel suport incondicional, pels tocs d'atenció quan ho mereixia, per contribuir en que em converteixi en la persona que sóc ara, per tenir tanta il·lusió com jo en que aquesta i totes les etapes que realitzi finalitzin amb èxit. Us estimo.

Gràcies de veritat,

Florencia Rímolo Figueira

Barcelona, 8 d'octubre de 2019

Glossari

AJAX: tecnologia que ens permet realitzar peticions a un servidor mitjançant JavaScript i obtenir la resposta en un segon pla.

API: de l'anglès *Application Programming Interface*, és un conjunt de mètodes i procediments que serveixen per ser utilitzats per un altre *software*.

CAS: de l'anglès *Central Authentication Service*, protocol SSO que permet a un usuari accedir a diverses aplicacions introduint les seves credencials un cop.

Depurador: eina que ens permet depurar, és a dir, buscar i corregir errors en el codi.

Framework: esquema o patró que ens permet estructurar millor el codi i estalviar temps a l'hora de desenvolupar.

Front-end: aquell codi o part de l'aplicació que és visible i entendible per l'ésser humà.

Mapejar: relacionar una entitat amb una altra.

Responsive: disseny d'aplicacions web que consisteix en adaptar l'aparença d'aquestes al dispositiu que s'estigui utilitzant per visitar-les.

SSO: de l'anglès *Single Sign-On*, és un sistema d'autenticació que permet a un usuari accedir a més d'un sistema identificant-se només una vegada.

Software de codi obert: aquell *software* on el seu codi font es publica pel seu ús o modificació pels desenvolupadors.

Software lliure: aquell *software* on el seu codi es pot utilitzar lliurement per modificar-lo o redistribuir-lo aportant millores.

Índex

Resum.....	2
Resumen.....	3
Abstract.....	4
Agraïments.....	5
Glossari.....	6
Índex	7
Índex d'imatges	11
Índex de taules.....	13
1. Introducció.....	15
2. Context.....	16
2.1. Formulació del problema.....	16
2.2. Actors implicats.....	18
2.2.1. Tema.....	18
2.2.2. Tecnologia.....	18
2.2.3. Ús.....	18
2.2.4. Desenvolupament.....	19
2.3. Estat de l'art.....	20
2.3.1. Aplicacions actuals	20
2.3.1.1. Aplicació de gestió de dades.....	20
2.3.1.2. Aplicació web de sol·licitud d'estància acadèmica	21
2.3.2. Aplicacions similars	23
2.3.2.1. Erasmus Without Paper	23
2.3.3. Interacció amb altres sistemes	24
2.3.3.1. API de la FIB.....	24
2.3.3.2. SSO UPC.....	25
2.3.4. Valoració de les alternatives.....	25
2.4. Visió del projecte.....	26
3. Abast del projecte i planificació temporal.....	29

3.1. Abast del projecte	29
3.1.1. Objectius del projecte	29
3.1.2. Delimitació de l'abast.....	29
3.1.3. Metodologia.....	30
3.1.4. Obstacles	31
3.2. Planificació inicial.....	31
3.2.1. Planificació general	31
3.2.2. Fases del projecte	32
3.2.2.1. Anàlisi	32
3.2.2.2. Planificació.....	32
3.2.2.3. Autoaprenentatge	32
3.2.2.4. Desenvolupament aplicació web	32
3.2.2.5. Documentació	33
3.2.2.6. Defensa	33
3.2.3. Estimació d'hores	33
3.2.4. Recursos requerits	34
3.2.4.1. Recursos humans	34
3.2.4.2. Recursos software	35
3.2.4.3. Recursos hardware	35
3.2.4.4. Altres recursos	35
3.2.4.5. Recursos per tasca	35
3.2.5. Diagrama de Gantt	37
3.2.6. Valoració d'alternatives i pla d'acció	39
3.3. Planificació final	39
3.3.1. Diagrama de Gantt final.....	40
3.3.2. Hores dedicades finals	43
4. Gestió econòmica.....	44
4.1. Identificació i estimació de costos	44
4.1.1. Costos directes.....	44
4.1.2. Costos indirectes	46

4.1.3. Imprevistos	47
4.1.4. Contingència	48
4.1.5. Pressupost total.....	48
4.2. Control de gestió.....	49
4.3. Pressupost final	49
5. Sostenibilitat i compromís social.....	52
5.1. Econòmica.....	52
5.2. Ambiental.....	52
5.3. Social.....	53
6. Especificació	54
6.1. Requisits funcionals	54
6.1.1. Diagrama del procés en BPMN	54
6.1.2. Esquema conceptual de dades.....	57
6.1.3. Diagrames d'estats.....	60
6.1.4. Diagrames de casos d'ús	63
6.1.5. <i>Brief style</i>	65
6.1.5.1. Administrador RRH.....	65
6.1.5.2. Administrador de la universitat <i>partner</i>	66
6.1.5.3. Estudiant.....	67
6.1.5.4. Sistema.....	67
6.1.6. Especificació completa dels casos d'ús	68
6.1.6.1. Cas d'ús (CU01): Editar sol·licitud.....	68
6.1.6.2. Cas d'ús (CU02): Crear nominació.....	69
6.1.6.3. Cas d'ús (CU03): Editar nominació	70
6.1.7. Històries d'usuari.....	71
6.1.7.1. Història d'usuari (HU01): Editar sol·licitud	71
6.1.7.2. Història d'usuari (HU02): Crear nominació	71
6.1.7.3. Història d'usuari (HU03): Editar nominació.....	72
6.2. Requisits no funcionals	74
6.2.1. Aparença.....	74

6.2.2. Facilitat d'ús	74
6.2.3. Personalització i internacionalització	75
6.2.4. Aprenentatge	75
6.2.5. Fiabilitat i disponibilitat	75
6.2.6. Interacció amb sistemes adjacents	76
6.2.7. Implantació	76
6.2.8. Accés	76
6.2.9. Privacitat	77
7. Disseny i implementació	78
7.1. Tecnologies utilitzades	78
7.1.1. Django	78
7.1.2. Bootstrap	78
7.1.3. jQuery	78
7.1.4. Oracle Database	79
7.1.5. PyCharm	79
7.1.6. DBeaver	79
7.1.7. Git	79
7.1.8. GitLab UPC	80
7.2. Arquitectura del projecte	80
7.3. Model-Vista-Controlador (MVC)	81
7.4. Estructura de la base de dades	82
7.5. Detalls de la implementació	85
7.5.1. Estructura del directori	85
7.5.2. Models	86
7.5.3. Controladors	87
7.5.4. Vistes	88
7.5.5. Comunicació amb l'API	89
7.5.6. <i>Login</i> amb l'SSO UPC	89
7.5.7. Registre d'usuaris	90
7.6. Proves	91

8. Posada en producció	95
8.1. Arquitectura	95
8.2. Implantació	96
9. Conclusions.....	98
10. Treball futur	99
11. Bibliografia.....	100
12. Annex.....	103
12.1. Diagrama del procés de l'aplicació actual	103
12.2. Prova de concepte EWP	104
12.2.1. Primers passos.....	104
12.2.2. Arquitectura de la xarxa	107
12.2.3. Com fer una petició	108
12.2.4. API Echo	109
12.2.5. API Institutions.....	110
12.3. Captures de pantalla de l'aplicació.....	112

Índex d'imatges

Imatge 1: Pantalla de l'aplicació actual pels incoming vista des de l'usuari administrador. Font: elaboració pròpia.	22
Imatge 2: API de la FIB. Font: elaboració pròpia.....	24
Imatge 3: SSO UPC. Font: elaboració pròpia.....	25
Imatge 4: Esquema del futur procés del sistema a desenvolupar. Font: elaboració pròpia.....	27
Imatge 5: Tasques del diagrama de Gantt inicial. Font: elaboració pròpia.	37
Imatge 6: Diagrama de Gantt inicial. Font: elaboració pròpia.	38
Imatge 7: Tasques del diagrama de Gantt final. Font: elaboració pròpia.....	40
Imatge 8: Diagrama de Gantt final. Font: elaboració pròpia.	41
Imatge 9: Hores reals dedicades de les tasques del projecte.....	43
Imatge 10: Diagrama del procés definitiu en BPMN. Font: elaboració pròpia.	55
Imatge 11: Esquema conceptual de les dades. Font: elaboració pròpia.....	57

Imatge 12: Diagrama d'estats d'una nominació. Font: elaboració pròpia.....	60
Imatge 13: Diagrama d'estats d'una sol·licitud. Font: elaboració pròpia.	61
Imatge 14: Usuaris del sistema. Font: elaboració pròpia.	63
Imatge 15: Diagrama de casos d'ús de l'usuari administrador. Font: elaboració pròpia.	63
Imatge 16: Diagrama de casos d'ús de l'usuari estudiant. Font: elaboració pròpia.....	64
Imatge 17: Diagrama de casos d'ús de l'usuari administrador de la universitat partner. Font: elaboració pròpia.	64
Imatge 18: Diagrama de casos d'ús del sistema. Font: elaboració pròpia.	65
Imatge 19: Esquema de l'arquitectura del projecte. Font: elaboració pròpia.....	80
Imatge 20: Esquema del patró arquitectònic MVC. Font: elaboració pròpia.	82
Imatge 21: Taules de la base de dades. Font: elaboració pròpia.	82
Imatge 22: Estructura del directori de l'aplicació. Font: elaboració pròpia.	85
Imatge 23: Un exemple de model de Django. Font: elaboració pròpia.	87
Imatge 24: Exemple de controlador a Django. Font: elaboració pròpia.	87
Imatge 25: Exemple de vista a Django. Font: elaboració pròpia.....	88
Imatge 26: Obtenció dels plans d'estudi a través de l'API de la FIB. Font: elaboració pròpia.....	89
Imatge 27: Configuració de l'SSO. Font: elaboració pròpia.	90
Imatge 28: Mapeig de les URLs de l'SSO. Font: elaboració pròpia.	90
Imatge 29: Prova unitària 1. Font: elaboració pròpia.	91
Imatge 30: Prova unitària 2. Font: elaboració pròpia.	92
Imatge 31: Prova unitària 3. Font: elaboració pròpia.	92
Imatge 32: Prova unitària 4. Font: elaboració pròpia.	93
Imatge 33: Prova unitària 5. Font: elaboració pròpia.	93
Imatge 34: Resultat de les proves unitàries. Font: elaboració pròpia.....	94
Imatge 35: Arquitectura del projecte a producció. Font: elaboració pròpia.	95
Imatge 36: Diagrama del procés en BPMN de l'aplicació actual. Font: elaboració pròpia.....	103
Imatge 37: Codi de l'arxiu manifest.xml. Font: elaboració pròpia.....	105
Imatge 38: Esquema de l'estructura general de la xarxa EWP. Font: https://github.com/erasmus-without-paper/ewp-specs-architecture	107
Imatge 39: Diagrama de seqüència d'una petició a l'API Registry. Font: elaboració pròpia.....	108

Imatge 40: Diagrama del disseny d'una petició a l'API Echo. Font: elaboració pròpia.	109
Imatge 41: Resposta de la petició a l'API Echo. Font: elaboració pròpia.	110
Imatge 42: Diagrama de seqüència d'una petició a l'API Institutions. Font: elaboració pròpia.	110
Imatge 43: Resposta de la petició a l'API Institutions. Font: elaboració pròpia.	111
Imatge 44: Pantalla formulari de nominació. Font: elaboració pròpia.	112
Imatge 45: Pantalla de nominacions i sol·licituds. Font: elaboració pròpia.	112
Imatge 46: Pantalla per editar les assignatures no ofertes. Font: elaboració pròpia.	113
Imatge 47: Pantalla amb resum d'una sol·licitud. Font: elaboració pròpia.	113
Imatge 48: Pantalla del formulari de sol·licitud (1). Font: elaboració pròpia.	114
Imatge 49: Pantalla del formulari de sol·licitud (2). Font: elaboració pròpia.	114
Imatge 50: Pantalla del formulari de sol·licitud (3). Font: elaboració pròpia.	115
Imatge 51: Pantalla del formulari de sol·licitud (4). Font: elaboració pròpia.	115
Imatge 52: Pantalla del formulari de sol·licitud (5). Font: elaboració pròpia.	116
Imatge 53: Pantalla del formulari de sol·licitud (6). Font: elaboració pròpia.	116
Imatge 54: Learning Agreement generat per l'aplicació. Font: elaboració pròpia.	117
Imatge 55: Carta d'acceptació generada per l'aplicació. Font: elaboració pròpia.	118

Índex de taules

Taula 1: Estimació inicial en hores de les tasques del projecte.	34
Taula 2: Recursos necessaris per cada tasca.	36
Taula 3: Estimació dels costos per activitat.	45
Taula 4: Estimació dels costos de recursos humans.	45
Taula 5: Estimació dels costos del recursos hardware i software.	46
Taula 6: Estimació d'altres costos indirectes.	47
Taula 7: Total de l'estimació de costos indirectes.	47
Taula 8: Estimació dels costos d'imprevistos.	48
Taula 9: Estimació dels costos de contingència.	48
Taula 10: Estimació del pressupost total.	49

Taula 11: Costos finals per activitat.....	50
Taula 12: Costos finals de recursos humans.....	50
Taula 13: Pressupost total final.	51

1. Introducció

Aquest Treball de Final de Grau (TFG) és un projecte realitzat a l'empresa l'inLab FIB [\[1\]](#) amb la col·laboració de l'Àrea de Relacions Internacionals (RRII) de la FIB, que és la unitat a la qual està destinat el resultat d'aquest projecte.

L'inLab FIB és un laboratori d'innovació i recerca de la Facultat d'Informàtica de Barcelona (UPC), especialitzat en aplicacions i serveis basats en les tecnologies més actuals. Aquesta empresa té més de 40 anys d'experiència col·laborant i desenvolupant projectes no només per la pròpia facultat (el Racó o la web de la FIB), sinó també destinats a entitats externes, tant públiques com privades.

Per la seva banda, l'Àrea de Relacions Internacionals de la FIB és la unitat que s'encarrega de gestionar tot el que té a veure amb la mobilitat acadèmica: estudiants, aliances internacionals, programes de mobilitat, etc. Aquesta unitat està formada pel vicedegà o vicedegana d'Estudiantat, que és el màxim responsable, i el personal administratiu.

La motivació d'aquest projecte és agilitzar i monitoritzar el procés de gestió dels estudiants d'universitats internacionals que volen realitzar una estància acadèmica de durada determinada a la FIB, de manera que es produeixi un estalvi del temps dedicat per l'Àrea de RRII i, en general, de totes les persones implicades en el procés.

Però aquest treball no només pretén facilitar i millorar la manera en que es realitzen les tasques relacionades amb la mobilitat internacional. També volem contribuir en cuidar el nostre planeta estalviant paper durant el procés d'admissió dels estudiants, ja que ara per ara es produeix un malbaratament innecessari de documentació en paper.

Actualment existeix un projecte europeu anomenat *Erasmus Without Paper* [\[2\]](#) que pretén eliminar la documentació en paper del procés de mobilitat acadèmica mitjançant la creació d'una xarxa de comunicació entre universitats en la qual l'intercanvi d'informació és purament digital. Veurem, doncs, en quin estat es troba aquest projecte i si és viable incorporar-lo en el nostre.

2. Context

2.1. Formulació del problema

Abans de definir el problema, explicarem alguns conceptes de l'àmbit de mobilitat acadèmica que són necessaris per entendre correctament el context del projecte.

- **Conveni bilateral:** acord entre dues universitats que estableix els requisits i condicions per tal que els estudiants puguin realitzar estades acadèmiques a la universitat estrangera i viceversa.
- **Incoming:** estudiant d'una altra universitat que realitza una estància de durada determinada (un quadrimestre o un any) a la FIB. Cal que existeixi un conveni bilateral amb la universitat de procedència.
- **Outgoing:** estudiant de la FIB que realitza una estància de durada determinada (un quadrimestre o un any) a una universitat amb la qual es té un conveni bilateral.
- **Nominació:** procediment que consisteix en la comunicació, per part d'una universitat, de les dades de l'estudiant que desitja realitzar una estància acadèmica a una altra universitat amb la qual té un acord bilateral. Les nominacions es poden realitzar únicament durant un període determinat, que depèn de cada universitat.
- **Universitat *partner*:** universitat amb la qual la FIB té signat un conveni bilateral.

Actualment, disposem d'acords bilaterals amb 197 universitats internacionals de 42 països [\[3\]](#). Per cada una d'aquestes universitats hi ha un procés de nominació pels estudiants de la FIB que volen realitzar una estància acadèmica a una d'aquestes institucions. Aquest procés es realitza segons el sistema que cada universitat hagi implementat:

- Per correu electrònic: s'envia a la direcció indicada al contracte tota la informació que es requereix sobre l'estudiant que es vol nominar.
- Aplicació web: algunes universitats implementen una aplicació web des de la qual s'accedeix amb les credencials proporcionades o unes pròpies per tal de nominar els estudiants.
- Formularis de Google: s'accedeix a l'enllaç proporcionat per la universitat en el qual s'omplen els diferents camps requerits amb les dades pertinents de l'estudiant.

En quant al procés de nominació dels estudiants *incoming*, ara per ara es realitza de forma manual per correu electrònic: concretament, la universitat *partner* envia a l'àrea de RRII la informació bàsica de l'estudiant que desitja realitzar un programa de mobilitat. A partir d'aquí s'iniciaria el procés de sol·licitud d'estància acadèmica: un cop l'estudiant ha estat nominat, accedeix a l'aplicació web actual de gestió de sol·licituds dels estudiants *incoming*, que consisteix en un formulari que ha d'omplir amb les dades necessàries per demanar una plaça a la FIB.

Un cop l'estudiant ha omplert aquest formulari, ha d'enviar per correu postal al vicedegà o vicedegana d'Estudiantat de la FIB documentació com el currículum, l'expedient acadèmic, el certificat d'idiomes, l'*Application Form*, que bàsicament són les dades del formulari que es generen en format PDF per poder imprimir-les, i la proposta de *Learning Agreement* (a partir d'ara LA). El LA és un document oficial que representa l'acord acadèmic per l'estància acadèmica que realitzarà l'estudiant. Concretament, conté les assignatures que vol cursar amb els crèdits, si vol realitzar el projecte final de carrera i dades de l'estància com el període i la titulació. Tota aquesta documentació ha d'estar degudament signada per l'estudiant sol·licitant i pel responsable acadèmic de la universitat origen.

Quan els documents arriben a la FIB, el personal administrador de l'Àrea de RRII han de revisar totes aquestes dades, així com el vicedegà o vicedegana d'Estudiantat de la FIB és qui ha de donar el vist-i-plau al LA provisional. Pot passar que la proposta de LA no sigui acceptada perquè les assignatures que hagi escollit l'estudiant no compleixen els requisits entre elles. En aquest cas, s'estableix una comunicació via *mail* entre l'Àrea de RRII i l'estudiant per arribar a un nou acord, que posteriorment ha de ser acceptat pel vicedegà o vicedegana d'Estudiantat de la FIB. Un cop acceptat, s'envia per correu postal a la universitat origen l'*Application Form* amb el *Learning Agreement* ja aprovat i degudament signat. Podem veure un diagrama del procés actual a l'apartat [12.1](#) (Annex).

Del procés de nominació i sol·licitud actual, podem extreure una sèrie d'inconvenients. Veiem que la feina de revisar manualment per correu totes les nominacions és una feina costosa, ja que resulta difícil gestionar totes les dades que es reben: concretament, cada curs acadèmic es reben entre 150 i 160 nominacions, de les quals acaben sent acceptats uns 120 estudiants. A més, el fet d'enviar per correu postal documentació com el seu expedient acadèmic o currículum, a part de l'impacte ambiental negatiu que genera el fet d'utilitzar paper, suposa haver d'esperar a que arribin correctament els documents i, en el cas que en falti algun o no sigui vàlid, s'haurien de tornar a enviar. En el nostre cas estaríem parlant d'uns 10 fulls per estudiant, 1600 fulls per curs en el pitjor cas, i això comporta un malbaratament de paper i temps innecessari avui en dia.

Un altre inconvenient és que l'eina actual per la gestió de dades està implementada amb una tecnologia obsoleta que dificulta el seu manteniment i actualització.

Així doncs, es requereix una solució que solventi aquests problemes de manera que faciliti les tasques a l'àrea de RRII i faciliti el manteniment a les persones que s'encarreguin del seu desenvolupament.

2.2. Actors implicats

A continuació es descriuen les parts interessades o *stakeholders* d'aquest projecte així com el rol que aplicaran al projecte i els objectius que tenen. Els separarem en 4 categories: tema, tecnologia, ús i desenvolupament.

2.2.1. Tema

- Expert del domini: Àrea de Relacions Internacionals

Rol: Proporcionar tots els seus coneixements sobre el procés d'admissió d'estudiants d'intercanvi com la documentació que cal aportar o la normativa.

Objectius:

- Assegurar que l'equip de desenvolupament té en compte tots els conceptes.

2.2.2. Tecnologia

- Director del projecte

Rol: Determinar quina arquitectura física tindrà el sistema (en quin servidor s'allotjarà, quina base de dades utilitzarà...).

Objectius:

- Que la implantació del projecte es realitzi de manera fàcilment mantenible.

- Desenvolupadora

Rol: Determinar el disseny de la base de dades i el *framework* que s'utilitzarà per implementar l'aplicació.

Objectius:

- Assegurar que l'estructura de la base de dades compleixi els estàndards de disseny.

2.2.3. Ús

- Àrea de Relacions Internacionals

Rol: Fer ús de l'aplicació d'acord amb les normes i els terminis de la facultat.

Objectius:

- Poder consultar tant les nominacions i les sol·licituds dels estudiants des de la mateixa aplicació.
 - Contactar amb els estudiants en cas que falti alguna dada en qualsevol moment.
 - Poder consultar les sol·licituds d'un període determinat.
 - Accedir a l'aplicació en qualsevol moment.
 - Poder ordenar les sol·licituds per diferents camps perquè sigui més fàcil la seva consulta.
- Universitats *partner*
Rol: Fer ús de l'aplicació per nominar els estudiants durant el termini de nominacions que s'hagi establert.
Objectius:
 - Nominar estudiants en qualsevol moment dintre del termini de nominacions.
 - Poder modificar les dades d'una nominació en cas que siguin errònies.
 - Estudiants *incoming*
Rol: Fer ús de l'aplicació per introduir les dades i documentació requerida per sol·licitar una estància acadèmica abans que finalitzi el termini establert.
 - Consultar l'estat de la seva sol·licitud un cop enviada en qualsevol moment.
 - Guardar i canviar les dades de la sol·licitud abans d'enviar-la.

2.2.4. Desenvolupament

- Equip de desenvolupament - Director del projecte
Rol: Organitzar, planificar i gestionar el correcte desenvolupament del projecte dins de la franja temporal designada per a la seva especificació, desenvolupament i posterior implantació.
Objectius:
 - Assegurar que la desenvolupadora realitza la feina que s'espera dins el termini establert.
- Equip de desenvolupament - Ponent del projecte
Rol: Donar suport acadèmic a la desenvolupadora durant tot el projecte
Objectius:
 - Assegurar que la memòria té una bona estructura i contingut.
 - Assegurar que es compleixen les competències transversals establertes a l'inici del projecte.
 - Que la memòria compleixi els estàndards de la facultat.

- Equip de desenvolupament - Desenvolupadora

Rol: Analitzar els requisits, determinar el model lògic de l'aplicació i realitzar la seva implementació i testeig. Redactar la memòria.

- Assegurar que el sistema compleixi els requisits establerts per l'Àrea de RRII.
- Implementar una aplicació que en un futur pugui ampliar-se amb noves funcionalitats.
- Que el codi estigui testejat i verificat amb jocs de proves al final de l'implementació.
- Complir els terminis establerts.
- Poder contactar en qualsevol moment amb l'Àrea de RRII per fer un seguiment del projecte i verificar que es van complint els requisits.
- Assegurar una bona qualitat de la memòria final.

- Àrea de Relacions Internacionals

Rol: Determinar què es vol i desitja de l'aplicació.

Objectius:

- Assegurar que el resultat final compleix les expectatives.

2.3. Estat de l'art

Actualment, no existeix una norma estàndard per les eines de gestió de mobilitat acadèmica. Per tant, cada universitat o institució fa ús de les aplicacions que consideri oportunes per realitzar aquesta tasca. A la FIB, disposem per una part d'una aplicació d'escriptori on es consulta i es gestiona tota la informació relacionada amb la mobilitat i per l'altra part d'una aplicació web on els estudiants nominats accedeixen per realitzar la sol·licitud d'estància acadèmica. A continuació es detallarà cada una d'aquestes eines.

2.3.1. Aplicacions actuals

2.3.1.1. Aplicació de gestió de dades

Per guardar i obtenir la informació necessària durant el procés de gestió de mobilitat, es fa servir una aplicació d'escriptori. Aquesta informació la podem classificar en diferents categories:

- **Universitats *partner*.** les universitats amb les quals tenim convenis bilaterals. Es guarden dades com el codi de la universitat, el nom, el país, la direcció i informació de contacte.

- **Estudiants *outgoing*:** dades personals dels estudiants que realitzen una estància acadèmica fora de la FIB, com el nom complet, document d'identitat, correu electrònic i universitat de procedència.
- **Estudiants *incoming*:** dades personals dels estudiants que realitzen l'estància acadèmica a la facultat, com el nom complet, document d'identitat, correu electrònic i universitat de procedència.
- **Convenis bilaterals:** es guarda l'estat dels convenis bilaterals amb les universitats, la data de caducitat del conveni i el nombre d'estudiants que es poden acceptar.

Amb aquesta informació es poden realitzar diversos tràmits. A continuació podem veure els més importants:

- Generar certificats, tant d'arribada com de sortida.
- Generar llistats d'estudiants, de correus electrònics, o d'universitats amb acords bilaterals.
- Assignar places a les universitats als estudiants *outgoing*. Aquesta funcionalitat és important ja que assignar places és un procés costós de realitzar manualment, degut a que s'han de tenir en compte diferents paràmetres (com el nombre de places disponibles de la universitat, el nivell d'idiomes requerit, el nivell de crèdits requerit, etc.) i, gràcies a aquesta aplicació es pot fer de forma semi-automàtica ja que té implementat un algorisme per realitzar aquesta tasca, tot i que posteriorment s'ha de revisar manualment si l'assignació ha estat correcta o no.
- Donar d'alta universitats i convenis bilaterals.

2.3.1.2. Aplicació web de sol·licitud d'estància acadèmica

Aquesta aplicació va ser desenvolupada fa uns anys com a projecte de Treball Final de Grau [\[4\]](#). Quan els estudiants són nominats, han de registrar-s'hi per poder sol·licitar una estància acadèmica introduint totes les dades que es requereixen que, posteriorment, seran revisades i validades per la FIB per tal d'acceptar o rebutjar la sol·licitud. Aquestes dades són:

- **Informació personal:** nom complet, document d'identitat, correu electrònic i data de naixement.

- **Informació acadèmica:** l'expedient acadèmic.
- **Dades de l'estància:** quin pla d'estudis vol cursar (Grau en Enginyeria Informàtica, Màster en Enginyeria Informàtica, Màster en Intel·ligència Artificial i Màster en Innovació i Recerca en Informàtica), quines assignatures vol fer (ja sigui del pla d'estudis seleccionat o d'un altre, sempre i quan no excedeixi el límit de crèdits requerits) o si vol realitzar el TFG. En aquest últim cas, s'haurà de trobar un projecte i un director.
- **Dades de la universitat *partner*:** concretament, les dades de contacte de l'Àrea de RRII de la universitat origen.
- **Nivell de llengües:** l'estudiant ha de proporcionar el certificat d'idiomes corresponent als requisits del pla que desitgi cursar.

A part d'aquesta informació, com s'ha comentat a l'apartat [2.1](#) hi ha documentació que actualment s'ha d'enviar per correu postal degudament signada tant per la universitat origen com per la universitat destí.

Name	Surname	Institution	Period&Program	State	Last access			
			-	Registered		Close	Edit	View
			-	Registered		Close	Edit	View
		F LYON12	1 - ERASMUS+	Dumped	14:44, 02-07	Return	Edit	
		CH LAUSANN06	A - ERASMUS+	Dumped	12:30, 10-06	Return	Edit	
		D MANNHEI01	2 - ERASMUS+	Dumped	13:41, 20-11	Return	Edit	
		PL KRAKOW02	2 - ERASMUS+	Dumped	08:56, 12-12	Return	Edit	
		D DARMSTA01	A - ERASMUS+	Dumped	11:16, 26-05	Return	Edit	
		TR ISTANBU20	2 - ERASMUS+	Open	11:54, 13-11	Close	Edit	View
		HR ZAGREB01	2 - ERASMUS+	Dumped	14:42, 16-11	Return	Edit	
		TR KAYSERIO4	1 -	Denied	13:23, 02-04	Return	Edit	
		I NAPOLI09	1 - ERASMUS+	Dumped	22:23, 29-11	Return	Edit	

Imatge 1: Pantalla de l'aplicació actual pels incoming vista des de l'usuari administrador. Font: elaboració pròpia.

Un cop enviada tota aquesta informació, es procedeix a la seva validació. A la imatge 1 es veu la pantalla a la qual accedeix l'Àrea de RRH, on poden veure les dades proporcionades per tots els estudiants per tal de consultar-les o editar-les i finalment acceptar-los com a estudiants *incoming* a la FIB.

2.3.2. Aplicacions similars

Tal i com s'ha comentat anteriorment, actualment no existeix una aplicació universal que interconnecti totes les institucions amb la informació dels seus respectius estudiants per gestionar el procés de mobilitat acadèmica. Així doncs, cada universitat implementa el mètode o *software* que millor s'adapti a les seves necessitats i infraestructura. Per exemple, d'una banda hi ha universitats on el procés de nominació es realitza per correu electrònic informant sobre les dades de l'estudiant seleccionat per realitzar l'estància i, d'altra banda, hi ha institucions que utilitzen un portal web on les universitats *partner* accedeixen per omplir un formulari de nominació amb la informació de l'estudiant, com és el cas de la Universidad Carlos III de Madrid [5]. En quant al procés de sol·licitud d'estàncies acadèmiques, cal destacar que al ser un procés que realitzen els propis estudiants, no tenim accés a les plataformes que fan servir les universitats i per tant, es desconeix quines eines fan servir per aquesta funció.

Existeix, però, una iniciativa europea que pretén digitalitzar tota la informació administrativa relacionada amb la mobilitat acadèmica mitjançant diferents protocols d'intercanvi de dades entre universitats. Aquest concepte reb el nom d'*Erasmus Without Paper* i el veurem a continuació.

2.3.2.1. Erasmus Without Paper

Erasmus Without Paper (a partir d'ara EWP) és un projecte europeu que vol posar fi a la documentació en paper durant el procés de gestió de mobilitat, d'aquí el seu nom. Cal dir que no és una aplicació com a tal, sinó que és una xarxa de protocols en la qual les diferents universitats es poden registrar i fer ús de les APIs que implementen aquests protocols per tal de comunicar-se entre elles, tot de manera purament digital.

EWP va ser presentat l'any 2017 [6] com a concepte en fase de proves que acaba madurant l'any 2018 amb una conferència el mes de desembre a Polònia [7] en la qual es presenta oficialment la versió 2.0, posant fi a la fase de *test* del projecte. En aquesta nova versió, es pot accedir al registre de la xarxa per poder veure quines universitats l'estan implementant [8]. Al ser un projecte molt

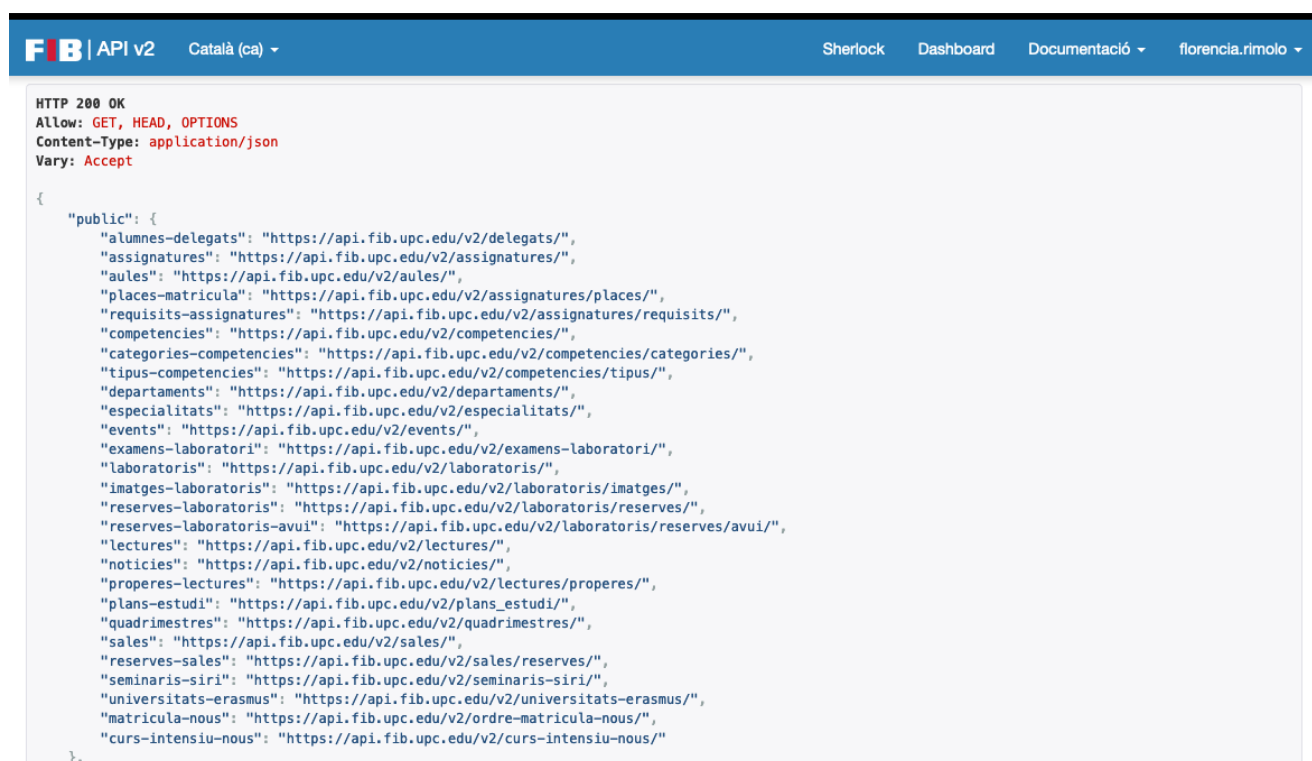
recent, es pot observar que no hi ha moltes institucions registrades i, per tant, això pot resultar un inconvenient de cara a incorporar-lo al nostre projecte.

2.3.3. Interacció amb altres sistemes

Existeixen altres sistemes que ens poden oferir funcionalitats per la nostra aplicació i, per tant, implementarem la interacció amb aquests per poder aprofitar seus serveis i d'aquesta manera evitar implementar funcionalitats que ja estan fetes.

2.3.3.1. API de la FIB

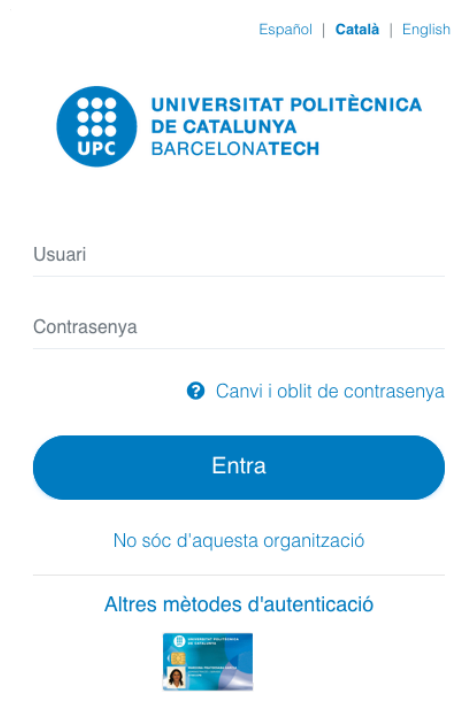
L'API de la FIB [9] és una aplicació web que ens proporciona accés a les dades públiques de la FIB en diferents formats (JSON, XML...), com per exemple els plans d'estudi o les assignatures. Així doncs, tota la informació que estigui disponible a l'API es consultarà des del nostre sistema de manera que no caldria emmagatzemar informació que ja ens ofereix aquesta aplicació.



Imatge 2: API de la FIB. Font: elaboració pròpia.

2.3.3.2. SSO UPC

L'SSO UPC [\[10\]](#) és un sistema d'autenticació de la UPC que implementa el protocol CAS. Aquest és el sistema que farem servir per autenticar als usuaris administradors de l'Àrea de Relacions Internacionals. D'aquesta manera no hauran de registrar-se al sistema, sinó que hi podran accedir directament amb les credencials de la UPC.



Imatge 3: SSO UPC. Font: elaboració pròpia.

2.3.4. Valoració de les alternatives

Després d'haver vist les aplicacions que es fan servir actualment i les aplicacions similars de les quals es poden treure idees, podem fer les següents consideracions:

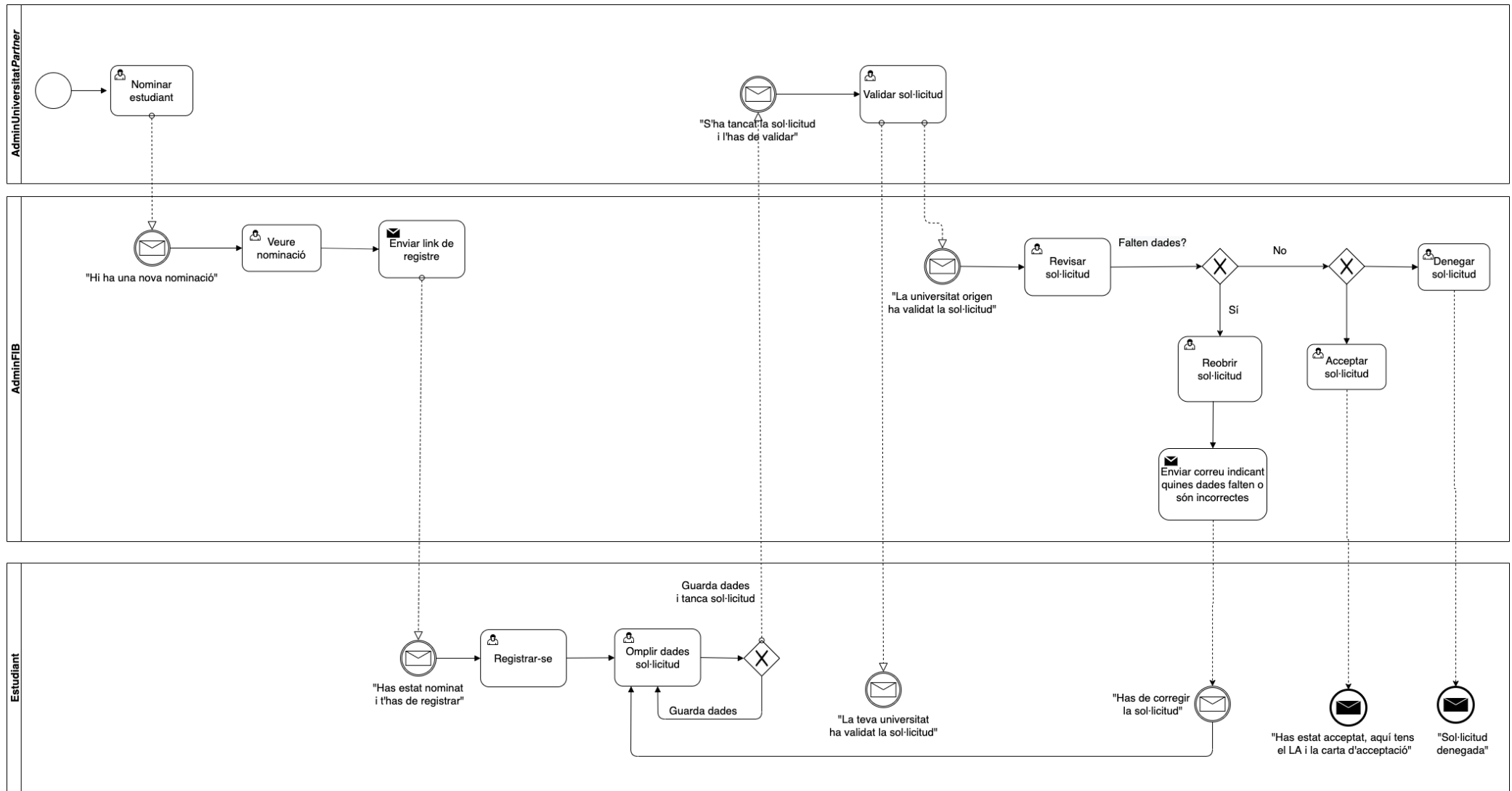
- L'aplicació d'escriptori de gestió de dades que es fa servir actualment, encara que tingui una tecnologia obsoleta, compleix correctament amb la seva funció i, a més, al ser una aplicació interna que només es fa servir des de l'Àrea de RRH, no hi ha necessitat de canviar-la per un altre. Per tant, no implementarem aquesta funcionalitat a la nostra aplicació, encara que es deixa oberta la seva possible incorporació en un futur.

- El projecte europeu *Erasmus Without Paper* és una bona idea per incorporar al nostre projecte en un futur, encara que hi ha poques universitats registrades degut a que és un concepte molt recent. Caldrà veure, doncs, com va evolucionant i madurant aquest concepte.
- No es coneixen les eines de gestió de mobilitat que implementen altres universitats, degut a que el seu accés està restringit als estudiants seleccionats i als propis administradors del sistema.
- L'aplicació web que es fa servir actualment té una tecnologia actualment obsoleta i, per tant, és difícil d'actualitzar i mantenir. A més, no s'hi poden pujar alguns documents requerits per realitzar una estància acadèmica, com el *Learning Agreement*, que s'han d'enviar per correu postal.

En conclusió, es pot dir que la millor alternativa possible és una nova aplicació web que incorpori una tecnologia actual per facilitar el seu manteniment, i que implementi tant el procés de nominacions com el de sol·licitud d'estàncies acadèmiques. A més, s'estudiarà en profunditat la implementació de la xarxa EWP, ja que considerem que és un projecte que pot aportar un valor a la nostra aplicació i, tot i que sigui un concepte molt recent, creiem que en un futur pot créixer i estendre's a totes o la gran majoria d'universitats europees.

2.4. Visió del projecte

Per tal de tenir una idea inicial del que ha de fer el sistema a desenvolupar, s'ha elaborat un esquema del nou procés de sol·licitud d'estàncies acadèmiques després de la primera reunió amb el client, d'aquesta manera tenim un punt de partida sobre el qual treballar. Aquest esquema anirà canviant durant el desenvolupament del projecte per anar adaptant-se el màxim possible a les necessitats del client .



Imatge 4: Esquema del futur procés del sistema a desenvolupar. Font: elaboració pròpia.

El procés s'inicia quan un administrador d'una universitat *partner* nomina un estudiant aportant les dades personals i acadèmiques necessàries. Aquesta nominació serà notificada a l'administrador de l'àrea de RRII de la FIB perquè la pugui revisar i informar a l'estudiant que ha estat nominat i s'ha de registrar al sistema per omplir les dades de la seva sol·licitud.

Un cop l'estudiant accedeix al sistema, ha d'omplir les dades i ha d'aportar la documentació necessària. Anteriorment, aquesta documentació l'havia d'enviar per correu postal a la FIB, però com s'ha comentat això genera un impacte ambiental negatiu i és per això que volem que l'estudiant pugui aportar aquests documents directament al sistema. Quan acabi d'omplir la sol·licitud, l'estudiant pot decidir si tancar-la definitivament perquè sigui revisada o, si encara no està segur que tot estigui correcte, guardar-la seguir editant-la més tard. En el primer cas, l'administrador de la universitat a la qual pertany l'estudiant serà notificat i informat que ha de validar la sol·licitud. Quan la validi, l'administrador de l'àrea de RRII de la FIB rebrà una notificació informant que ja pot revisar les dades. En el cas que no falti cap dada ni cap documentació, es decidirà si aquella sol·licitud ha estat acceptada o no. En cas contrari, s'haurà de comunicar a l'estudiant que ha de tornar a modificar alguna dada de la sol·licitud.

El procés finalitza quan l'estudiant hagi estat o no acceptat. Si ha estat admès a la facultat s'enviarà per correu electrònic el *Learning Agreement* i la carta d'acceptació degudament signats pel vicedegà o la vicedegana de la FIB. Això últim també és un pas endavant en quant a l'estalvi de paper, ja que anteriorment aquests documents s'enviaven per correu postal.

3. Abast del projecte i planificació temporal

A partir de la idea inicial que hem definit sobre el nou sistema es delimitarà l'abast del projecte i es realitzarà la planificació temporal.

3.1. Abast del projecte

3.1.1. Objectius del projecte

L'objectiu principal d'aquest projecte és construir un sistema *software* que automatitzi el major nombre de subprocessos possible dins el procés general de sol·licitud d'estàncies acadèmiques explicat a l'apartat [2.1](#), de manera que ajudi i faciliti a la realització de les tasques tant a l'Àrea de RRII, com als estudiants i a les universitats *partner*.

Aquest objectiu principal és molt general i poc mesurable, per la qual cosa hem definit una sèrie de sub-objectius que, assolits en conjunt, impliquen el compliment del gran objectiu esmentat.

- Construir un sistema *software* que elimini al 100% el treball en paper. Sabem la importància de l'estalvi de paper per contribuir en un planeta sostenible i per aquesta raó és imprescindible un sistema que no impliqui cap intercanvi de paper.
- Construir un sistema *software* que mantingui informats als estudiants en tot moment de l'estat de la seva sol·licitud d'estància acadèmica. Amb l'aplicació actual els estudiants han d'esperar a que la seva documentació enviada per correu postal arribi correctament, de manera que desconeixen en quin estat es troba la seva sol·licitud fins el moment. Volem un sistema que informi immediatament sobre qualsevol canvi a les sol·licituds dels estudiants.
- Construir una prova de concepte de la xarxa EWP que serveixi com a referència en cas que en un futur es decideixi implementar alguna de les seves APIs.

3.1.2. Delimitació de l'abast

Abans de començar a dissenyar i implementar, és important tenir clar i definir què farà exactament el projecte per delimitar el seu context.

Per tal de complir exitosament l'objectiu comentat a l'apartat [3.1.1](#), s'han estudiat les aplicacions i fonts de dades que actualment es fan servir des de l'àrea de RRII amb la finalitat de trobar quines són les mancances que tenen i quins són els punts de millora, i així tenir un punt de partida sobre el qual començar a treballar.

A partir d'aquests punts, es vol implementar una aplicació web en la qual les universitats *partner* puguin nominar estudiants que, posteriorment, es registraran al sistema i proporcionaran tota la informació requerida, inclosa la documentació que actualment no es pot introduir al mateix portal com el *Learning Agreement*, que haurà de ser revisada per l'Àrea de Relacions Internacionals per poder ser admesos com a estudiants *incoming* a la FIB. Aquesta aplicació web haurà de comptar amb una tecnologia actual que permeti el seu manteniment.

A més, es vol implementar una prova de concepte de la xarxa EWP, que en un futur pugui servir com a referència per poder implementar qualsevol de les funcionalitats que ofereix aquesta xarxa.

Quedarà fora de l'abast, doncs, tot el que té a veure amb els estudiants *outgoing*, que tenen una gestió diferent a la dels estudiants que realitzen l'estància acadèmica a la FIB, tal i com s'ha explicat a l'apartat [2.1](#).

3.1.3. Metodologia

Degut a que el temps de què es disposa per desenvolupar aquest projecte és limitat, s'ha de planificar de forma eficient tota la feina que es vagi realitzant. En aquest cas, s'ha optat per una metodologia àgil basada en petites iteracions o cicles de desenvolupament, amb l'objectiu d'anar revisant periòdicament el resultat del producte, mitjançant reunions amb el client almenys uns dos cops al mes, per tal de corregir els possibles problemes que puguin sorgir i evitar presentar un resultat final que no sigui el desitjat pel client. També es realitzaran reunions periòdiques amb la ponent del projecte, la Maria Ribera Sancho-Samso, per realitzar un seguiment tant de la documentació com del propi desenvolupament del projecte.

Amb la finalitat de tenir un control de les tasques que es realitzen durant el desenvolupament del projecte, utilitzarem l'eina Trello [\[11\]](#), una aplicació web per administrar projectes amb la que ja hem treballat en assignatures de l'especialitat d'Enginyeria del Software. A més, s'utilitzarà l'eina Git [\[12\]](#) per tenir un control de les versions del codi font de l'aplicació.

3.1.4. Obstacles

Durant les diferents etapes del projecte, es poden trobar diversos obstacles o dificultats que s'han de conèixer a priori per poder afrontar els problemes que en derivin. A continuació es detallaran:

- **Gestió del temps:** com tots els projectes, hi ha un calendari que limita el seu temps de realització. Per tant, és necessari tenir-ho en compte i dur a terme una bona planificació del temps necessari per cada etapa del projecte per tal d'arribar als resultats esperats.
- **Canvis en l'especificació de requisits:** durant el projecte, pot passar o bé que el client ja no desitgi incorporar alguna determinada funcionalitat definida al principi, o bé que en desitgi incloure. Nosaltres, seguint els principis de l'*Agile Manifesto* [\[13\]](#), que són 12 principis universals que haurien de complir els projectes de metodologia àgil, ho hem de tenir en compte i hem d'estar oberts a qualsevol canvi, ja que el resultat del projecte va destinat al client, que serà l'usuari final de l'aplicació i qui esperi que es compleixin els objectius fixats.

3.2. Planificació inicial

Com s'ha comentat en l'anterior apartat, és important realitzar una bona gestió del temps per tal d'assolir amb èxit els objectius plantejats. En aquest apartat descriurem quina és la planificació temporal del projecte que es seguirà per arribar a la nostra meta, concretament detallarem quines fases es realitzaran i farem una estimació en hores. A més, es determinaran quins recursos són necessaris pel projecte i, finalment, valorarem les conseqüències que poden tenir els diferents obstacles que poden sorgir durant el seu desenvolupament.

3.2.1. Planificació general

La durada aproximada d'aquest TFG és de 8 mesos, concretament des de mitjans de febrer fins a finals d'octubre de 2019. El primer mes es destinarà a la gestió del projecte realitzada a l'assignatura de GEP, que correspon amb les etapes d'anàlisi i planificació del projecte, i posteriorment s'iniciarà la fase de desenvolupament, que finalitza amb l'entrega de la memòria i la defensa del TFG.

3.2.2. Fases del projecte

3.2.2.1. Anàlisi

Aquesta primera fase tracta sobre analitzar les necessitats dels usuaris finals del projecte per definir els objectius que ha de complir el resultat del mateix. Durant aquesta fase es realitzaran i definiran els següents aspectes:

- Reunió inicial amb el client
- Esquema del procés de gestió dels estudiants *incoming*
- Reunió inicial amb la ponent
- Abast del projecte
- Contextualització (quin problema hem de resoldre, parts interessades, objectius, estat de l'art)

3.2.2.2. Planificació

La fase de planificació també forma part de la gestió del projecte i té a veure amb l'estructuració de les fases del projecte, el càlcul del pressupost i l'elaboració d'un informe de sostenibilitat.

Concretament:

- Planificació temporal
- Gestió econòmica i sostenibilitat

3.2.2.3. Autoaprenentatge

Durant aquesta fase, s'estudiarà el funcionament de les aplicacions que es fan servir actualment i de la xarxa de protocols *Erasmus Without Paper*.

3.2.2.4. Desenvolupament aplicació web

Aquesta fase és la més llarga, ja que és l'etapa en la qual elaborem l'aplicació web que serà el resultat d'aquest projecte. Aquí és on entra la metodologia àgil: cada setmana, concretament un dia a determinar, hi haurà una reunió entre la desenvolupadora i el director del projecte per planificar les tasques a dur a terme durant la setmana i per revisar la feina realitzada la setmana anterior. També es realitzaran reunions amb el personal de l'Àrea de RRII cada dues o tres setmanes, segons

la rellevància de les funcionalitats implementades fins al moment, i amb la ponent del projecte, com a mínim dos cops al mes.

Així doncs, no tindrem fases d'especificació de requisits, disseny, implementació, proves i implantació clarament diferenciades com en altres tipus de projectes *software*, sinó que aquestes fases es realitzaran de forma iterativa durant tot el projecte. Cal dir, però, que la durada de les iteracions no està prèviament definida, sinó que variarà segons la granularitat de les tasques que es planifiquen a les reunions. Per aquesta raó quan es faci el càlcul d'estimació d'hores, no dividirem el desenvolupament en subfases.

3.2.2.5. Documentació

En aquesta etapa es documentarà la memòria del projecte que es presentarà abans de la seva defensa. Cal dir que no és una fase que es realitzi després d'una altra, sinó que es va realitzant durant tot el projecte per tal de minimitzar la càrrega de treball a la fase final del mateix. La fase de documentació finalitzarà amb l'entrega final de la memòria del projecte.

3.2.2.6. Defensa

Amb aquesta fase finalitza el projecte. Consisteix en defensar el projecte davant d'un Tribunal de TFG i en presentar el resultat final del projecte al client.

3.2.3. Estimació d'hores

En la següent taula es pot veure el temps aproximat en hores que portarà realitzar cada una de les tasques descrites a l'apartat anterior. Per calcular aquesta estimació, s'ha tingut en compte l'horari laborable que és de 20h a la setmana i el període de vacances d'estiu durant tot l'agost.

Cal destacar que les tasques d'abast, contextualització i totes les tasques de la fase de planificació, formen part de la documentació i, per tant, no es sumaran les hores dedicades a aquestes tasques a l'hora d'estimar les hores de la fase de documentació. Aquestes últimes s'han estimat suposant que es dediquen 3h a la setmana a documentar.

	TASCA	DEDICACIÓ (HORES)	Predecessores
1	Anàlisi	32	
2	Reunió inicial amb el client	2	
3	Esquema procés gestió dels <i>incoming</i>	4	2
4	Reunió inicial amb la ponent	1	3
5	Abast	10	4
6	Contextualització	15	4
7	Planificació	24	1
8	Planificació temporal	10	
9	Gestió econòmica i sostenibilitat	14	8
10	Autoaprenentatge	56	4
11	Estudi aplicacions actuals	24	
12	Estudi EWP	32	
13	Desenvolupament de l'aplicació	600	10
14	Desenvolupament de l'aplicació	495	
15	Reunions	105	
16	Documentació	80	4
17	Defensa	33	16
18	Preparació de la defensa	30	
19	Defensa del projecte	1	
20	Presentació client	2	
21	TOTAL	825	

Taula 1: Estimació inicial en hores de les tasques del projecte.

3.2.4. Recursos requerits

En aquest apartat llistarem els recursos que es requereixen tant de forma directa com indirecta al nostre projecte, ja que al següent apartat de gestió econòmica els haurem de tenir en compte per estimar el pressupost.

3.2.4.1. Recursos humans

R1. Director del projecte (2h a la setmana)

R2. Ponent del projecte (1h cada dues setmanes)

R3. Personal de l'Àrea de Relacions Internacionals de la FIB (2h cada dues setmanes)

R4. Desenvolupadora

3.2.4.2. Recursos software

R5. Mozilla Thunderbird 60.5: per comunicar-se amb les persones implicades en el projecte.

R6. Google Drive: per redactar la memòria.

R7. PyCharm 2018.3.5: per la implementació de l'aplicació web.

R8. Git 2.16.1: control de versions del codi de l'aplicació web.

R9. Trello: per la organització i control de les tasques del projecte.

R10. Base de Dades de la FIB: per guardar i consultar les dades de mobilitat acadèmica.

3.2.4.3. Recursos hardware

R11. Ordinador de sobretaula Dell Optiplex 7020 i7 a 3.60 GHz, 8 GB de RAM¹ i Windows 10: pel desenvolupament de l'aplicació web i la memòria.

R12. Ordinador portàtil MacBook Air i5 a 1.8 GHz, 8 GB de RAM i macOS Mojave: per redactar la memòria.

R13. Servidor base de dades: allotja la base de dades.

R14. Servidor producció: allotjarà l'aplicació web testejada i completament funcional.

3.2.4.4. Altres recursos

R15. Internet

R16. Electricitat

R17. Transport

3.2.4.5. Recursos per tasca

A la següent taula es mostren els recursos necessaris per cada tasca. Els recursos d'internet, electricitat i transport (R15, R16 i R17) han estat necessaris durant tot el projecte i no els posarem a la taula per tal de no sobrecarregar-la.

	TASCA	RECURSOS
1	Anàlisi	R1-R4, R6, R11, R12
2	Reunió inicial amb el client	R1, R3, R4
3	Esquema procés gestió dels <i>incoming</i>	R4, R6, R11
4	Reunió inicial amb la ponent	R1, R2, R4
5	Abast	R1, R4, R6, R11, R12
6	Contextualització	R1, R4, R6, R11, R12
7	Planificació	
8	Planificació temporal	R1, R4, R6, R11, R12
9	Gestió econòmica i sostenibilitat	R1, R4, R6, R11, R12
10	Autoaprenentatge	R4, R11
11	Estudi aplicacions actuals	R4, R11
12	Estudi EWP	R4, R11
13	Desenvolupament de l'aplicació	R1-R11, R13, R14
14	Desenvolupament de l'aplicació	R1, R4, R5-R11, R13, R14
15	Reunions	R1, R2, R3, R4
16	Documentació	R12
17	Defensa	R1-R4, R6
18	Preparació de la defensa	R4, R6
19	Defensa del projecte	R1, R2, R4
20	Presentació client	R1, R3, R4

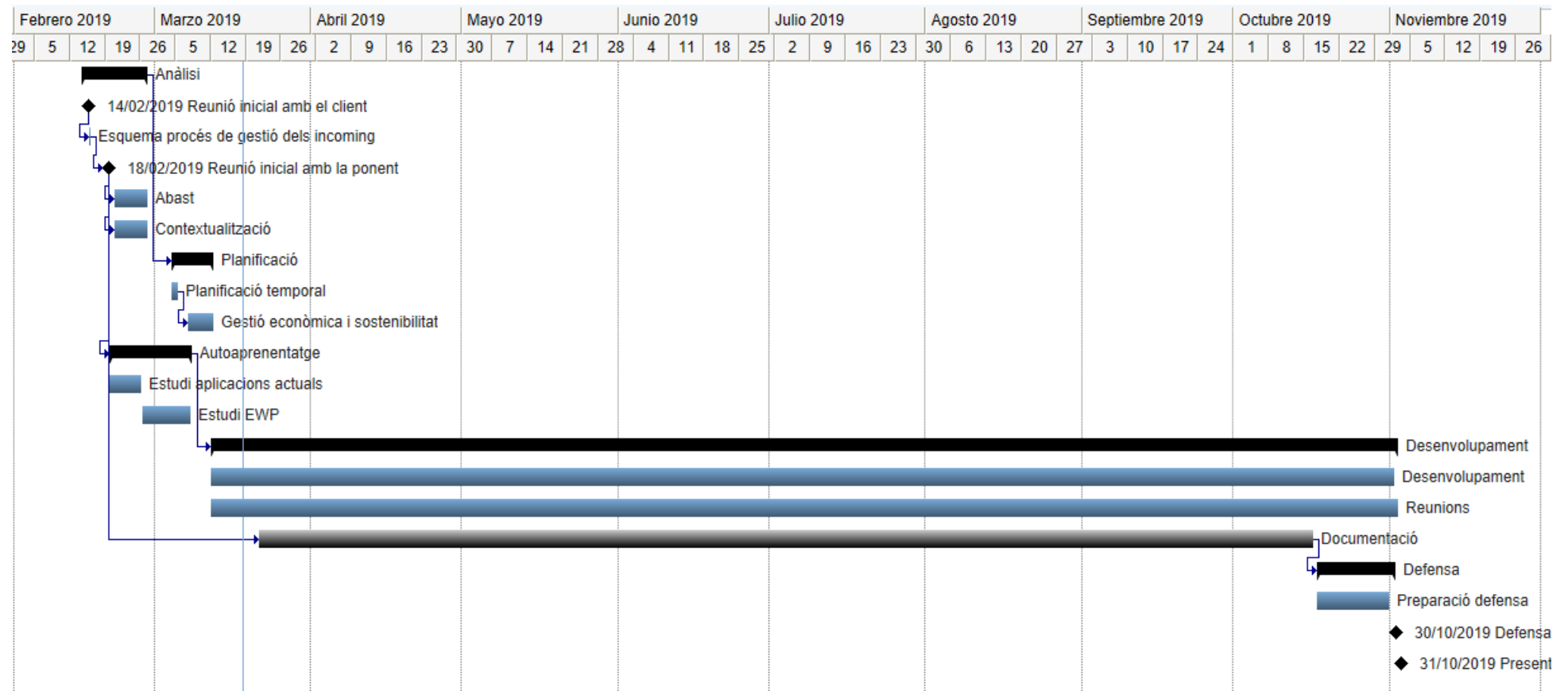
Taula 2: Recursos necessaris per cada tasca.

3.2.5. Diagrama de Gantt

A continuació es mostren les tasques i el diagrama de Gantt realitzats a l'inici del projecte:

	Nombre	Inicio	Fin	Predecessoras
1	Anàlisi	14/02/2019	26/02/2019	
2	Reunió inicial amb el client	14/02/2019	14/02/2019	
3	Esquema procés de gestió dels incoming	15/02/2019	15/02/2019	2
4	Reunió inicial amb la ponent	17/02/2019	18/02/2019	3
5	Abast	20/02/2019	26/02/2019	4
6	Contextualització	20/02/2019	26/02/2019	4
7	Planificació	03/03/2019	11/03/2019	1
8	Planificació temporal	03/03/2019	05/03/2019	
9	Gestió econòmica i sostenibilitat	07/03/2019	11/03/2019	8
10	Autoaprenentatge	19/02/2019	07/03/2019	4
11	Estudi aplicacions actuals	19/02/2019	25/02/2019	
12	Estudi EWP	26/02/2019	07/03/2019	
13	Desenvolupament	11/03/2019	31/10/2019	10
14	Desenvolupament	11/03/2019	31/10/2019	
15	Reunions	11/03/2019	31/10/2019	
16	Documentació	21/03/2019	15/10/2019	4
17	Defensa	15/10/2019	31/10/2019	16
18	Preparació defensa	15/10/2019	29/10/2019	
19	Defensa	30/10/2019	30/10/2019	
20	Presentació client	31/10/2019	31/10/2019	

Imatge 5: Tasques del diagrama de Gantt inicial. Font: elaboració pròpia.



Imatge 6: Diagrama de Gantt inicial. Font: elaboració pròpia.

3.2.6. Valoració d'alternatives i pla d'acció

Com s'ha comentat a l'apartat [3.1.4](#), pot passar que el projecte pateixi alguna modificació respecte a la planificació inicial. A continuació comentarem quines conseqüències tindrien aquests obstacles:

- **Una mala gestió del temps:** això pot tenir un greu impacte en la planificació temporal i per tant és molt important tenir-ho en compte. D'una banda, pot passar que s'hagin dedicat menys hores de les necessàries a la dedicació de la memòria. Això és un problema greu ja que no es pot defensar el projecte sense haver entregat la memòria. Per aquest motiu, la redacció s'anirà realitzant paral·lelament al desenvolupament del projecte, així evitem una gran càrrega de treball durant les últimes setmanes del TFG abans de l'entrega final. D'altra banda, pot passar que no s'hagi acabat la implementació del projecte a la data prevista de finalització. En aquest cas, tindríem un augment en el cost del projecte ja que s'hauran de dedicar hores extra per finalitzar-lo amb èxit. Suposarem que aquestes hores extra seran el 10% de les hores totals del projecte.
- **Canvis en els requisits:** pot passar que, durant el desenvolupament del projecte, el client desitgi afegir o eliminar alguna funcionalitat de l'aplicació web. En aquest cas, l'impacte seria mínim ja que es realitzaria el canvi el més aviat possible i abans de començar qualsevol altre funcionalitat.
- **Avaria de l'ordinador de sobretaula, del portàtil o dels servidors:** aquest obstacle no s'havia definit prèviament, ja que no havíem vist els recursos materials necessaris. Però s'ha de tenir en compte que tot equip informàtic es pot espatllar, i això generarà unes conseqüències. En aquest cas, les conseqüències serien un augment en el cost del projecte ja que hauríem de reparar o canviar la màquina avariada.

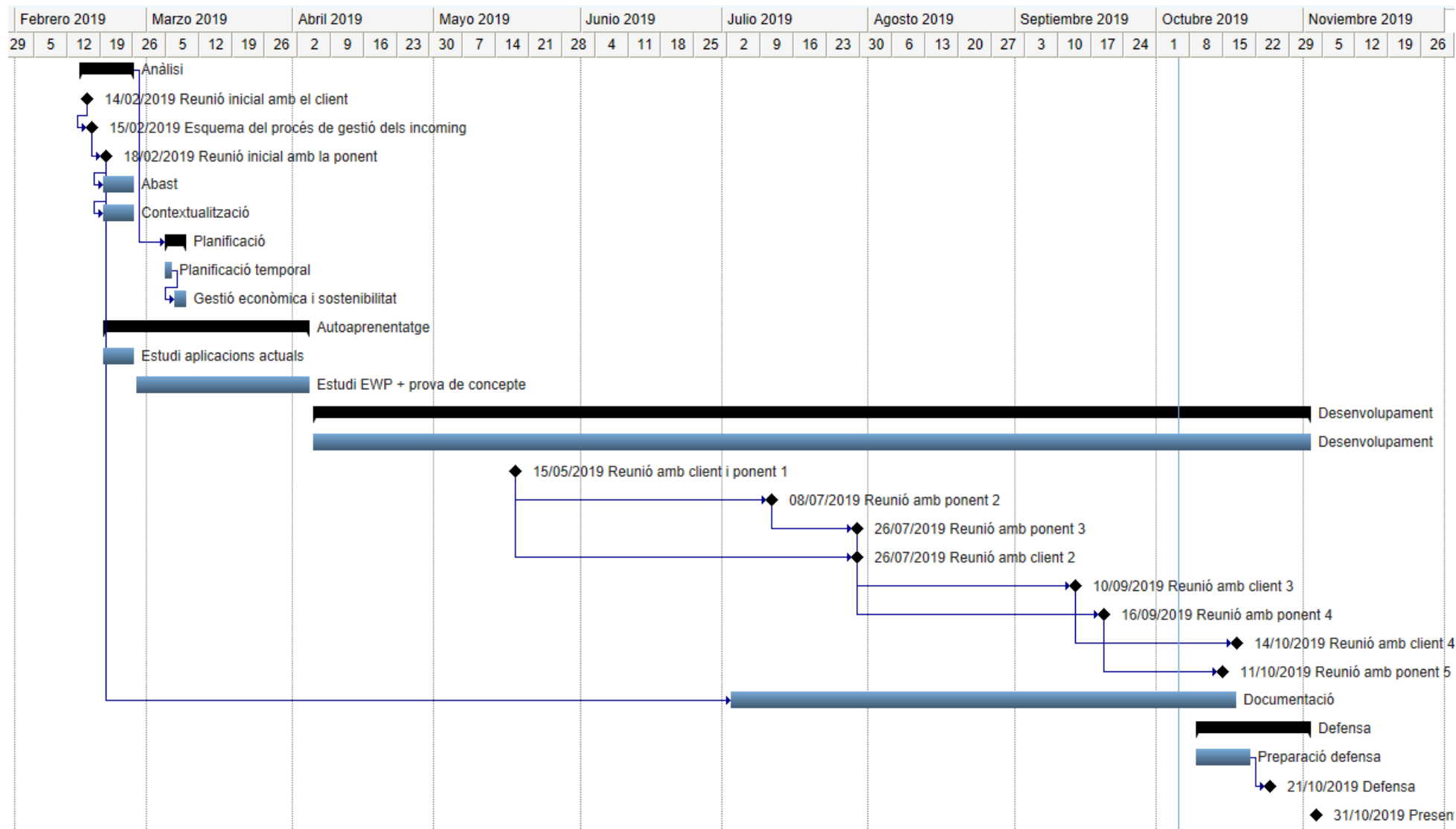
3.3. Planificació final

La planificació inicial és una estimació de les hores dedicades a cada fase del projecte. Això vol dir que es poden produir desviacions a mesura que es va desenvolupant, ja sigui perquè per algun motiu el projecte s'hagi iniciat més tard, o alguna fase hagi tingut una durada diferent de la que s'esperava. En aquest apartat veurem quines han estat aquestes desviacions mitjançant un diagrama de Gantt que mostra el temps real dedicat a cada fase.

3.3.1. Diagrama de Gantt final

	Nombre	Inicio	Fin	Predecessoras
1	<input checked="" type="checkbox"/> Anàlisi	14/02/2019	25/02/2019	
2	Reunió inicial amb el client	14/02/2019	14/02/2019	
3	Esquema del procés de gestió dels incoming	15/02/2019	15/02/2019	2
4	Reunió inicial amb la ponent	18/02/2019	18/02/2019	3
5	Abast	19/02/2019	25/02/2019	4
6	Contextualització	19/02/2019	25/02/2019	4
7	<input checked="" type="checkbox"/> Planificació	04/03/2019	08/03/2019	1
8	Planificació temporal	04/03/2019	05/03/2019	
9	Gestió econòmica i sostenibilitat	06/03/2019	08/03/2019	8
10	<input checked="" type="checkbox"/> Autoaprenentatge	19/02/2019	03/04/2019	
11	Estudi aplicacions actuals	19/02/2019	25/02/2019	
12	Estudi EWP + prova de concepte	26/02/2019	03/04/2019	
13	<input checked="" type="checkbox"/> Desenvolupament	04/04/2019	31/10/2019	
14	Desenvolupament	04/04/2019	31/10/2019	
15	Reunió amb client i ponent 1	15/05/2019	15/05/2019	
16	Reunió amb ponent 2	08/07/2019	08/07/2019	15
17	Reunió amb ponent 3	26/07/2019	26/07/2019	16
18	Reunió amb client 2	26/07/2019	26/07/2019	15
19	Reunió amb client 3	10/09/2019	10/09/2019	18
20	Reunió amb ponent 4	16/09/2019	16/09/2019	17
21	Reunió amb client 4	14/10/2019	14/10/2019	19
22	Reunió amb ponent 5	11/10/2019	11/10/2019	20
23	Documentació	01/07/2019	15/10/2019	4
24	<input checked="" type="checkbox"/> Defensa	07/10/2019	31/10/2019	
25	Preparació defensa	07/10/2019	18/10/2019	
26	Defensa	21/10/2019	21/10/2019	25
27	Presentació client	31/10/2019	31/10/2019	

Imatge 7: Tasques del diagrama de Gantt final. Font: elaboració pròpia.



Imatge 8: Diagrama de Gantt final. Font: elaboració pròpia.

Com es pot veure, la desviació més important és la durada de la fase d'autoaprenentatge, que ha suposat que l'etapa de desenvolupament s'hagi iniciat un mes més tard. Això és degut a que s'han hagut de dedicar més hores a estudiar i entendre el funcionament de la xarxa EWP, que ha resultat ser més complexa del que esperàvem. Però, malgrat aquesta desviació important, el desenvolupament del projecte no s'ha vist afectat negativament ja que el temps real dedicat ha estat suficient per complir les expectatives del projecte.

També podem observar que s'han afegit tasques, concretament les dates de les reunions amb el client i la ponent, ja que inicialment no es sabia amb exactitud quan es realitzarien aquestes reunions. Observem que a la planificació inicial vam plantejar que hi hauria cada setmana una reunió amb el director del projecte. Això finalment no ha estat així degut a que desenvolupadora i director treballen en el mateix lloc físic i la comunicació amb el director ha estat immediata en cada moment en que s'ha necessitat, i per tant no s'ha requerit cap reunió específica. A la planificació inicial, a més, es va comentar que es realitzarien reunions cada dues o tres setmanes amb el client i dues vegades al mes amb la ponent. Finalment no s'ha pogut complir aquesta periodicitat, d'una banda degut a la dificultat que ha suposat coincidir en disponibilitat per fer les reunions i d'altra banda degut a que s'ha preferit convocar reunions quan realment hi havia dubtes que frenessin la implementació d'alguna funcionalitat o quan ja teníem funcionalitat completes que es podien ensenyar. No obstant, el fet que no s'hagin pogut realitzar tantes reunions com teníem planificat ha suposat tenir més temps pel desenvolupament del projecte per la qual cosa compensa que aquesta fase s'hagi iniciat un mes més tard, com hem comentat.

3.3.2. Hores dedicades finals

	TASCA	DEDICACIÓ (HORES)	Predecessores
1	Anàlisi	32	
2	Reunió inicial amb el client	2	
3	Esquema procés gestió dels <i>incoming</i>	4	2
4	Reunió inicial amb la ponent	1	3
5	Abast	10	4
6	Contextualització	15	4
7	Planificació	24	1
8	Planificació temporal	10	
9	Gestió econòmica i sostenibilitat	14	8
10	Autoaprenentatge	128	4
11	Estudi aplicacions actuals	20	
12	Estudi EWP	108	
13	Desenvolupament de l'aplicació	538	10
14	Desenvolupament de l'aplicació	521	
15	Reunions	17	
16	Documentació	80	4
17	Defensa	33	16
18	Preparació de la defensa	30	
19	Defensa del projecte	1	
20	Presentació client	2	
21	TOTAL	835	

Imatge 9: Hores reals dedicades de les tasques del projecte.

Com es pot veure, malgrat les desviacions comentades anteriorment, el càlcul final de les hores de dedicades ha estat molt ben aproximat: només 10 hores més de les estimades.

4. Gestió econòmica

Un cop realitzada la planificació del projecte cal fer-ne un estudi econòmic. Per fer aquest estudi, es calcularà el pressupost del projecte mitjançant una estimació de costos dels recursos necessaris per la seva realització. També s'hauran de tenir en compte els possibles imprevistos que poden fer que els costos augmentin.

4.1. Identificació i estimació de costos

Per fer un càlcul estimat dels costos del projecte, cal diferenciar els costos directes que venen donats per les activitats del projecte, dels costos indirectes.

4.1.1. Costos directes

Per calcular els costos directes, hem de tenir en compte els recursos humans. En aquest cas es comptarà solament amb la desenvolupadora ja que és la única persona que realitzarà el projecte i que per tant realitzarà tots els rols. Es farà el càlcul en dues variants: la primera, tenint en compte que es tracta d'un projecte universitari i el preu per hora correspon al d'un becari (8€/h) [\[14\]](#) i, la segona, tindrà en compte la remuneració real que li correspon a cada rol segons l'estudi de remuneració de l'any 2018 de Page Personnel [\[15\]](#). En ambdós casos, el salari és brut. Suposarem que es treballa 251 dies a l'any.

A continuació es mostra la taula amb el càlcul dels costos per cada activitat realitzada al projecte, en la qual tindrem en compte l'impost de la Seguretat Social, que a la variant 1 serà del 2% i a la variant 2 del 34%. També s'ha detallat el cost de les tasques corresponents a la gestió del projecte.

Fase	Rol	Dedicació (h)	Cost per hora (€/h) (Variant 1)	Cost per hora (€/h) (Variant 2)	Total (€) (Variant 1)	Total (€) (Variant 2)
Anàlisi	Cap de projecte	32	8	25	256	800
Abast	Cap de projecte	10	8	25	80	250
Contextualització	Cap de projecte	15	8	25	120	375
Planificació	Cap de projecte	24	8	25	192	600
Planificació temporal	Cap de projecte	10	8	25	80	250
Gestió econòmica i sostenibilitat	Cap de projecte	14	8	25	112	350
Autoaprenentatge	Programadora	56	8	15	448	840
Desenvolupament de l'aplicació	Analista (10%), Dissenyadora (20%), Programadora (70%)	495	8	25, 10 i 15, respectivament	3.960	7.425
Reunions	Cap de projecte	105	8	25	840	2.625
Documentació	Cap de projecte	80	0	0	0	0
Defensa	Cap de projecte	33	8	25	264	825
TOTAL (sense SS)	-	825	-	-	5.960	13.115
TOTAL (amb SS)	-	825	-	-	7.152	17.574,10

Taula 3: Estimació dels costos per activitat.

D'aquesta taula, també podem extreure els costos dels recursos humans:

Rol	Dedicació (h)	Cost per hora (€/h) (Variant 1)	Cost per hora (€/h) (Variant 2)	Total (€) (Variant 1)	Total (€) (Variant 2)
Cap de projecte	194	8	25	1.552	4.850
Cap de projecte (fora de l'horari laboral)	80	0	0	0	0
Analista	49,5	8	25	396	1.237,50
Dissenyadora	99	8	10	792	990
Programadora	402,5	8	15	3.220	6.037,50
TOTAL (sense SS)	825	-	-	5.960	13.115
TOTAL (amb SS)	825	-	-	7.152	17.574,10

Taula 4: Estimació dels costos de recursos humans.

4.1.2. Costos indirectes

Els costos indirectes són aquells que no es produeixen directament pel fet de realitzar les tasques del projecte. Per calcular aquests costos, tindrem en compte els recursos *hardware*, els recursos *software* i altres costos com els desplaçaments a la universitat, l'electricitat o l'internet.

En quant als recursos *hardware* i *software* s'ha de tenir en compte la seva vida útil i per tant la seva amortització. Hisenda permet amortitzar un equip informàtic en 3-4 anys, de manera que si algun equip té més de 4 anys ja no computarà com a cost. L'amortització dels recursos *software*, en canvi, serà de 2-3 anys.

L'amortització es calcularà mitjançant la següent fórmula:

$$\frac{\text{Preu de compra de l'equip (€)} * \text{Hores ús equip TFG}}{\text{Vida útil equip anys (3 o 4)} * \text{Dies feiners any (220)} * \text{Hores treball dia (8)}}$$

Producte	Preu (€)	Temps dedicació TFG	Preu amortitzat (€)
Mozilla Thunderbird 60.5	0	8 mesos	0
Google Drive	0	8 mesos	0
PyCharm 2018.3.5	0	8 mesos	0
Git 2.16.1	0	8 mesos	0
Trello	0	8 mesos	0
Base de dades de la FIB	0	8 mesos	0
MacBook Air i5	1.360	80 h	15,46
Dell Optiplex 7020 i7	800	8 mesos	93,75
Servidor base de dades	1.500	8 mesos	175,78
Servidor producció	1.500	8 mesos	175,78
TOTAL	0	-	460,77

Taula 5: Estimació dels costos del recursos hardware i software.

Producte	Preu per unitat	Quantitat	Preu total (€)
Transport (T-Jove)	105 € / 3 mesos	3 unitats	315
Electricitat	0,12 € / kWh	0,5 kWh * 825 h	49,50
Internet	50 € / mes	8 mesos	400
TOTAL	-	-	764,50

Taula 6: Estimació d'altres costos indirectes.

En total, els costos indirectes seran:

Producte	Preu (€)
Mozilla Thunderbird 60.5	0
Google Drive	0
PyCharm 2018.3.5	0
Git 2.16.1	0
Trello	0
Base de dades de la FIB	0
MacBook Air i5	15,46
Dell Optiplex 7020 i7	93,75
Servidor base de dades	175,78
Servidor producció	175,78
Transport (T-Jove)	315
Electricitat	49,50
Internet	400
TOTAL	1.225,47

Taula 7: Total de l'estimació de costos indirectes.

4.1.3. Imprevistos

En tot projecte poden sorgir imprevistos que farien que el seu cost augmenti. En aquest cas, pot passar que l'ordinador de sobretaula, el portàtil o els servidors s'espatllin i, per tant, s'hauria de reparar o en el pitjor cas comprar-ne un de nou. Així doncs, com a màxim el cost del projecte augmentaria en el cost de l'ordinador a canviar. La probabilitat que això succeeixi és del 5%. També podria passar que no haguem realitzat correctament la planificació temporal del projecte i, en conseqüència, s'hagi de dedicar més temps per finalitzar-lo, per exemple el 10% de les hores totals

del projecte. Suposarem que seran hores de programació. La probabilitat que això succeeixi és del 10%.

Imprevist	Probabilitat	Cost (€) (Variant 1)	Cost (€) (Variant 2)
Avaria ordinador sobretaula	5 %	800 aproximadament	800 aproximadament
Avaria ordinador portàtil	5 %	1.360 en cas que se n'hagi de comprar un de nou	1.360 en cas que se n'hagi de comprar un de nou
Avaria servidor base de dades	5%	1.500 en cas que se n'hagi de comprar un de nou	1.500 en cas que se n'hagi de comprar un de nou
Avaria servidor producció	5%	1.500 en cas que se n'hagi de comprar un de nou	1.500 en cas que se n'hagi de comprar un de nou
Mala gestió del temps	10%	$0,1 * 825 \text{ h} * 8 \text{ €} / \text{h} = 660$	$0,1 * 825 \text{ h} * 15 \text{ €} / \text{h} = 1.237,50$
TOTAL	-	5.820	6.397,50

Taula 8: Estimació dels costos d'imprevistos.

4.1.4. Contingència

Inclourem un import en el pressupost com a pla de contingència, que calcularem a partir de la suma dels costos d'imprevistos amb la seva probabilitat:

Concepte	Probabilitat	Cost (€) (Variant 1)	Cost (€) (Variant 2)	Exposició al risc (€) (Variant 1)	Exposició al risc (€) (Variant 2)
Avaria ordinador sobretaula	5%	800	800	40	40
Avaria ordinador portàtil	5%	1.360	1.360	68	68
Avaria servidor base de dades	5%	1.500	1.500	75	75
Avaria servidor producció	5%	1.500	1.500	75	75
Mala gestió del temps	10%	660	1.237,50	66	123,75
TOTAL	-	-	-	324	381,75

Taula 9: Estimació dels costos de contingència.

4.1.5. Pressupost total

Sumant tots els costos calculats anteriorment obtenim el pressupost total del projecte:

Concepte	Cost (€) (Variant 1)	Cost (€) (Variant 2)
Costos directes	7.152	17.574,10
Costos indirectes	1.225,47	1.225,47
Imprevistos	5.820	6.397,50
Contingència	174	231,75
TOTAL	14.371,47	25.428,82

Taula 10: Estimació del pressupost total.

4.2. Control de gestió

Per tal de realitzar un control de gestió es farà un seguiment de les hores reals invertides a cada fase del projecte, de manera que un cop finalitzat es sabrà amb exactitud quantes hores s'hi han dedicat i, en cas de desviació, podrem calcular l'augment del cost del projecte si aquestes hores superen les previstes o dedicarem el temps en ampliar les funcionalitats de l'aplicació, en cas que el projecte s'acabi abans del temps previst. A més, es realitzarà un control dels recursos utilitzats tenint en compte les hores reals de consum.

En cas que el pressupost inicialment calculat s'hagi quedat curt, tindrem com a recurs el pla de contingència.

Per calcular aquestes desviacions podem utilitzar els següents indicadors:

- Desviació en la realització de tasques (cost): $(\text{cost estimat} - \text{cost real}) * \text{consum hores real}$
- Desviació d'un recurs *hardware* (cost): $(\text{cost estimat} - \text{consum real}) * \text{cost real}$
- Desviació en la realització de tasques (hores): $(\text{consum estimat} - \text{consum real}) * \text{cost real}$
- Desviació total en la realització de tasques: $\text{cost estimat total} - \text{cost real total}$
- Desviació total de recursos: $\text{cost estimat total} - \text{cost real total}$

4.3. Pressupost final

En aquest apartat calcularem els costos a partir de la planificació final per obtenir el pressupost final. Farem servir les mateixes taules que els apartats anteriors.

Fase	Rol	Dedicació (h)	Cost per hora (€/h) (Variant 1)	Cost per hora (€/h) (Variant 2)	Total (€) (Variant 1)	Total (€) (Variant 2)
Anàlisi	Cap de projecte	32	8	25	256	800
Abast	Cap de projecte	10	8	25	80	250
Contextualització	Cap de projecte	15	8	25	120	375
Planificació	Cap de projecte	24	8	25	192	600
Planificació temporal	Cap de projecte	10	8	25	80	250
Gestió econòmica i sostenibilitat	Cap de projecte	14	8	25	112	350
Autoaprenentatge	Programadora	128	8	15	1.024	1.920
Desenvolupament de l'aplicació	Analista (10%), Dissenyadora (20%), Programadora (70%)	521	8	25, 10 i 15, respectivament	4.168	7.815
Reunions	Cap de projecte	17	8	25	136	425
Documentació	Cap de projecte	80	0	0	0	0
Defensa	Cap de projecte	33	8	25	264	825
TOTAL (sense SS)	-	835	-	-	6.040	12.385
TOTAL (amb SS)	-	835	-	-	7.248	16.595,90

Taula 11: Costos finals per activitat.

Rol	Dedicació (h)	Cost per hora (€/h) (Variant 1)	Cost per hora (€/h) (Variant 2)	Total (€) (Variant 1)	Total (€) (Variant 2)
Cap de projecte	106	8	25	848	2.650
Cap de projecte (fora de l'horari laboral)	80	0	0	0	0
Analista	52,1	8	25	416,80	1.302,50
Dissenyadora	104,2	8	10	833,60	1.042
Programadora	492,7	8	15	3.941,60	7.390,50
TOTAL (sense SS)	835	-	-	6.040	12.3855
TOTAL (amb SS)	835	-	-	7.248	16.595,90

Taula 12: Costos finals de recursos humans.

Concepte	Cost (€) (Variant 1)	Cost (€) (Variant 2)
Costos directes	7.248	16.595,90
Costos indirectes	1.225,47	1.225,47
TOTAL	8.473,47	17.821,37

Taula 13: Pressupost total final.

Com podem veure, els costos s'han reduït considerablement respecte els que havíem estimat inicialment (-5.898€ en la variant 1 i -7.607,45€ en la variant 2). Això és degut a que no ha sorgit cap imprevist dels que havíem calculat, només s'ha produït un augment de 10h en les hores totals del projecte que ha provocat que augmentin en 96€ (variant 1) els costos per activitat.

Els costos indirectes no s'han hagut de tornar a calcular ja que aquests tenen en compte els recursos requerits, que no han canviat.

5. Sostenibilitat i compromís social

En aquest apartat s'ha realitzat un estudi de la sostenibilitat del projecte per determinar si el projecte és viable econòmica, social i ambientalment.

5.1. Econòmica

Per determinar si el projecte és viable econòmicament s'ha fet un càlcul del pressupost del projecte, en el qual s'ha tingut en compte els recursos humans i materials, els possibles imprevistos i contingències. Aquest projecte no serà comercialitzat sinó que serà exclusiu per l'Àrea de Relacions Internacionals de la FIB, per la qual cosa no es traurà cap benefici econòmic i en conseqüència no és econòmicament viable.

En quant a la sostenibilitat econòmica, observem que es podria millorar reduint el temps de realització: o bé contractant més personal o bé augmentant les hores de dedicació a jornades completes. No obstant, la primera solució suposaria un augment en el cost i la segona suposaria dedicar les mateixes hores totals a realitzar el projecte però en comptes de 20h a la setmana 40h, i això no seria possible degut als altres compromisos que té la desenvolupadora.

5.2. Ambiental

Els recursos necessaris per realitzar aquest projecte ja s'utilitzaven per fer altres tasques i, per tant, no caldrà comprar-ne de nous sinó que es reutilitzaran els ja existents. El portàtil i l'ordinador de sobretaula, que són els principals recursos que es faran servir, tenen un consum elèctric que s'ha de tenir en compte ja que afecta directament al medi ambient. No obstant això, el portàtil s'utilitza per realitzar altres tasques a part del projecte, per la qual cosa en aquest cas no hi haurà un augment significatiu del consum elèctric.

A més, amb el nou sistema no caldria fer ús de paper durant el procés de gestió dels estudiants *incoming*, de manera que això suposa una millora ambiental respecte a les eines que es fan servir actualment. De fet, com s'ha comentat a l'apartat [2.1](#), ens estalviariem uns 1.600 fulls de paper respecte l'aplicació actual ja que l'intercanvi d'informació i de dades en el sistema desenvolupat és purament digital.

Considerem que no es podria realitzar amb menys recursos dels que s'han utilitzat, ja que s'han fet servir els recursos mínims necessaris pel desenvolupament del projecte.

5.3. Social

Aquest projecte és el primer projecte real en el qual participo des del seu inici. Això suposa un repte molt gran per mi que em suposarà aprendre nous conceptes que podré aplicar en el meu futur professional.

En quant a l'impacte en la societat, al ser un projecte destinat a la facultat no en tindrà un impacte considerable. Qui es beneficiarà principalment serà l'Àrea de Relacions Internacionals, que és l'òrgan dintre de la FIB que va comunicar la seva necessitat de tenir una eina nova per gestionar els estudiants *incoming*, degut a les mancances de l'aplicació que es fa servir actualment. Així doncs, la solució d'aquest projecte els facilitarà aquesta tasca. A més, les universitats *partner* notaran una millora considerable amb el nou sistema de nominacions, que ja no es farà per correu electrònic sinó que es farà a través de la nostra aplicació web. També notaran una millora els estudiants que sol·liciten una estància acadèmica, ja que sabran en tot moment en quin estat es troba la seva sol·licitud degut a que no han d'esperar en cap moment que la documentació arribi correctament per correu postal, sinó que al realitzar-se digitalment aquesta arribarà d'immediat.

6. Especificació

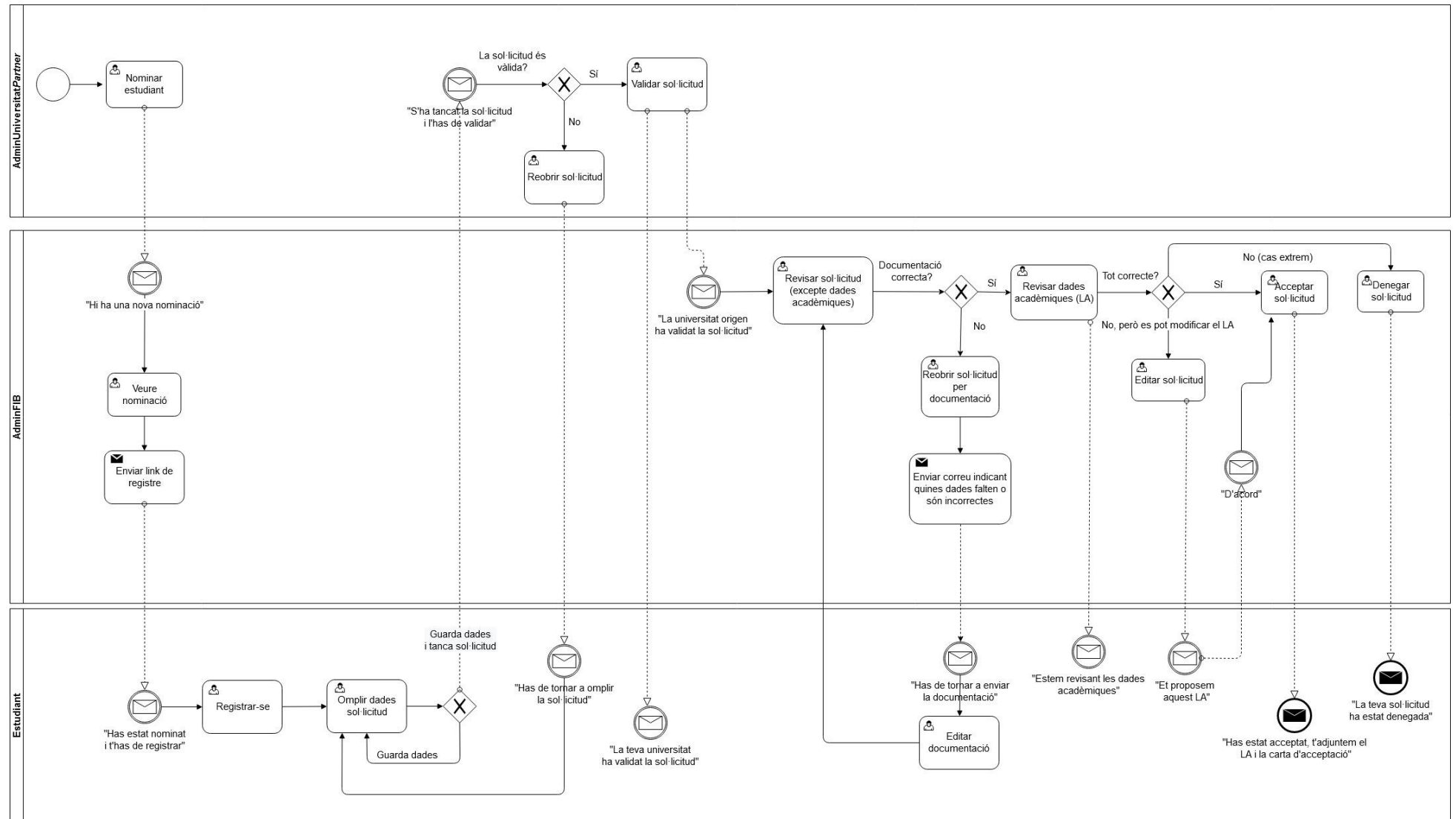
Durant aquesta fase definirem i detallarem els requisits del sistema que ens demana el client per arribar a la solució desitjada. En aquest cas, el client és l'Àrea de Relacions Internacionals de la FIB, amb qui hem realitzat diverses reunions durant tot el desenvolupament del projecte per tal d'anar adaptant el producte a les seves necessitats.

Els requisits que definirem es classifiquen en requisits funcionals i no funcionals.

6.1. Requisits funcionals

6.1.1. Diagrama del procés en BPMN

A l'apartat [2.4](#) s'ha elaborat un esquema inicial de com seria el nou procés de gestió dels estudiants *incoming*. Aquest esquema s'ha anat modificant durant les reunions amb el client adaptant-lo a les seves necessitats. A continuació podem veure l'esquema final del procés modelat en BPMN.



Imatge 10: Diagrama del procés definitiu en BPMN. Font: elaboració pròpia.

Els canvis s'han produït a partir del tancament de la sol·licitud de l'estudiant. Quan vam elaborar l'esquema inicial, no vam definir d'una manera clara el pas de la validació de la sol·licitud per part de la universitat origen. Aquest concepte es va anar definint durant les reunions, fins a tenir el diagrama final.

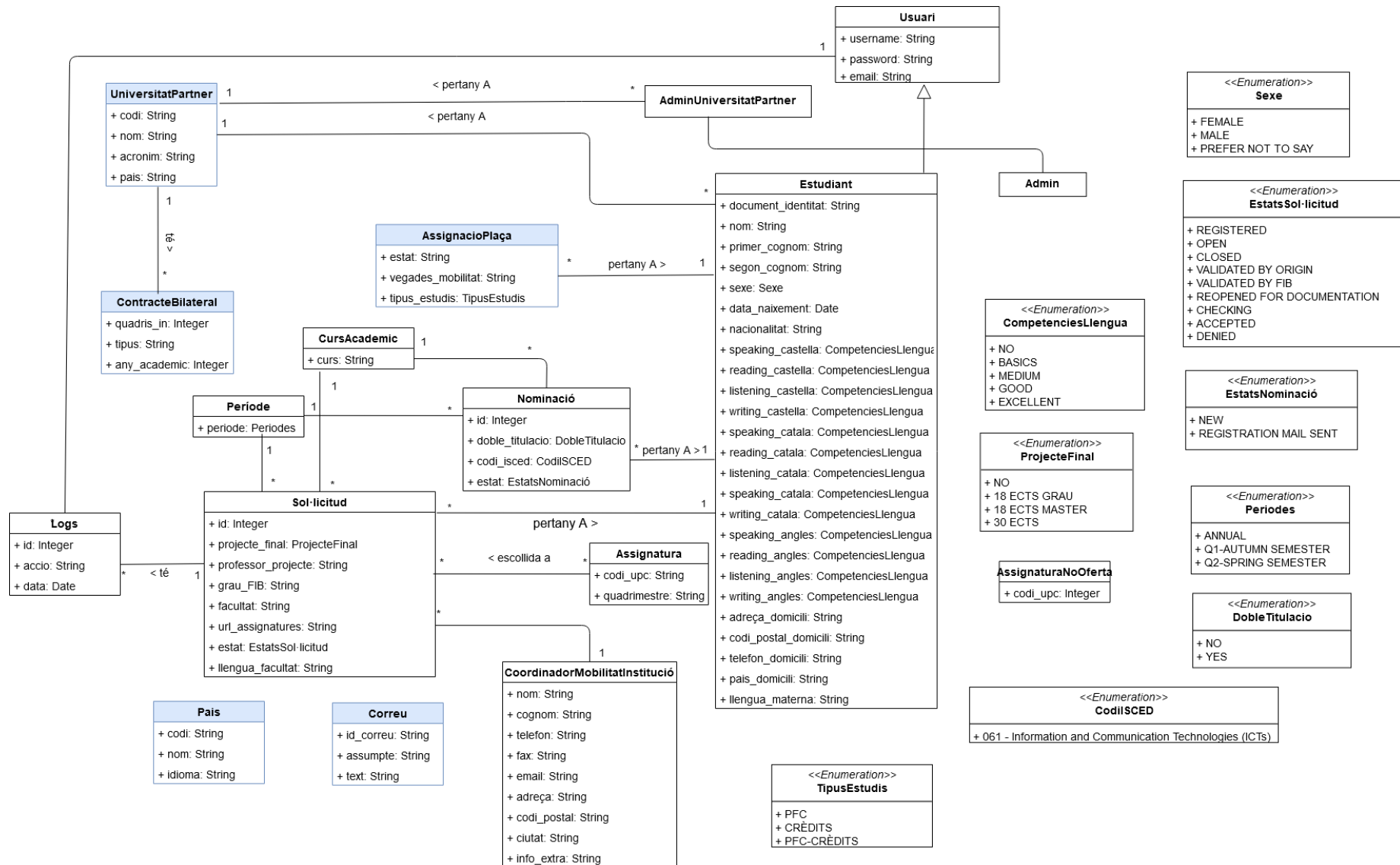
El pas de la validació de la universitat origen correspon al pas de signar l'*Application Form* i el LA provisional del procés de l'aplicació actual, de manera que és una tasca important. Per aquesta raó, si la universitat origen considera que la sol·licitud té dades errònies o que les assignatures que ha escollit l'estudiant no seran reconegudes, es posarà en contacte amb l'estudiant informant-lo que ha d'editar la seva sol·licitud.

En cas que la universitat origen validi la sol·licitud, l'àrea de RRII serà notificada i revisarà les dades i la documentació aportada per l'estudiant (excepte les dades acadèmiques). Si aquesta documentació no és la correcta, l'estudiant haurà de tornar-la a enviar, sense que la universitat origen hagi de tornar a validar la sol·licitud.

Un cop validada per part de la FIB, es revisen les dades acadèmiques, concretament les assignatures. Aquest pas seria l'equivalent a revisar el LA provisional del procés de l'aplicació actual. Si les assignatures que ha escollit l'estudiant compleixen els requisits, aquest és acceptat i rebrà el LA i la carta d'acceptació amb la signatura del vicedegà o la vicedegana d'Estudiantat. En cas contrari, s'inicia una negociació entre l'àrea de RRII i l'estudiant en la qual acorden un nou LA.

Veiem doncs que el pas de revisar la sol·licitud definit a la visió s'ha dividit en dos passos: la validació de la sol·licitud per part de la FIB i la revisió de les dades acadèmiques. Això s'ha fet d'aquesta manera perquè el personal administrador de l'àrea de RRII no revisa les assignatures escollides pels estudiants fins que la documentació requerida no estigui correcta.

6.1.2. Esquema conceptual de dades



Imatge 11: Esquema conceptual de les dades. Font: elaboració pròpia.

Restriccions textuais:

- Claus externes: (Usuari, username); (UniversitatPartner, codi); (Logs, id); (Nominació, id); (Assignatura, codi_upc); (CursAcademic, curs); (Periode, període); (Correu, id_correu) (Pais, codi+idioma); (Sol·licitud, id)
- Una UniversitatPartner no pot tenir més d'un contracte bilateral que finalitzi el mateix any acadèmic.
- Per una Nominació i una Sol·licitud que pertanyin al mateix estudiant, el CursAcadèmic i el Període han de ser els mateixos.
- Per un Estudiant, no hi pot haver una Sol·licitud sense que hi hagi una Nominació.
- Si el Període d'una Sol·licitud no és Anual, les Assignatures escollides han de tenir com a quadrimestre el mateix període.
- Per un CursAcadèmic i un Període, un Estudiant no pot tenir més d'una Nominació.

Cal destacar que hi ha dades que gestionarà l'aplicació directament i dades que nosaltres només consultarem, que són les que estan en color blau, ja que ja es feien servir a l'aplicació d'escriptori de gestió dels *incoming* i, per tant, ens serveixen per la nova aplicació.

En quant als usuaris, observem que tenim tres tipus:

- **Estudiant:** com el seu nom indica, són els estudiants que volen realitzar una estància acadèmica a la FIB i, per tant, seran aquells usuaris que aportaran les dades necessàries per sol·licitar-la.
- **AdminUniversitatPartner:** aquest tipus d'usuari el formaran els responsables de les universitats *partner*, que nominaran els estudiants.
- **Admin:** seran els administradors de l'aplicació, en el nostre cas els responsables de l'Àrea de RRII de la FIB.

Un estudiant és nominat per realitzar una estància acadèmica durant un curs i un període determinat, que pot ser tot el curs (anual), o un quadrimestre (Q1 o Q2). El nombre d'estudiants que

pot nominar una universitat ve condicionat pel nombre de quadrimestres que hi ha a l'acord bilateral, que en aquest cas correspon a l'atribut *quadris_in* de la classe *ContracteBilateral*. Posem com a exemple que aquest val 4, de manera que es poden donar els següents casos:

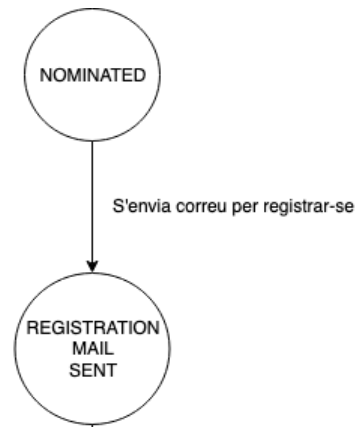
- La universitat nomina 2 estudiants on cada un d'ells vol realitzar una estància anual.
- La universitat nomina 3 estudiants, on 2 d'ells volen realitzar una estància d'un quadrimestre i l'altre estudiant una anual.
- La universitat nomina 4 estudiants, cada un d'ells per realitzar una estància d'un quadrimestre.

A la nominació tenim l'atribut doble titulació que indica si l'estudiant vol cursar una doble titulació durant l'estància, i també el codi ISCED (*International Standard Classification of Education*), que és un codi internacional desenvolupat per la UNESCO per classificar els diferents programes i nivells educatius.

A la sol·licitud tenim informació com la titulació que cursarà l'estudiant, la URL de la guia docent de les assignatures que ha cursat a la seva universitat o el tipus de projecte de final de carrera que vol, en cas que el vulgui realitzar.

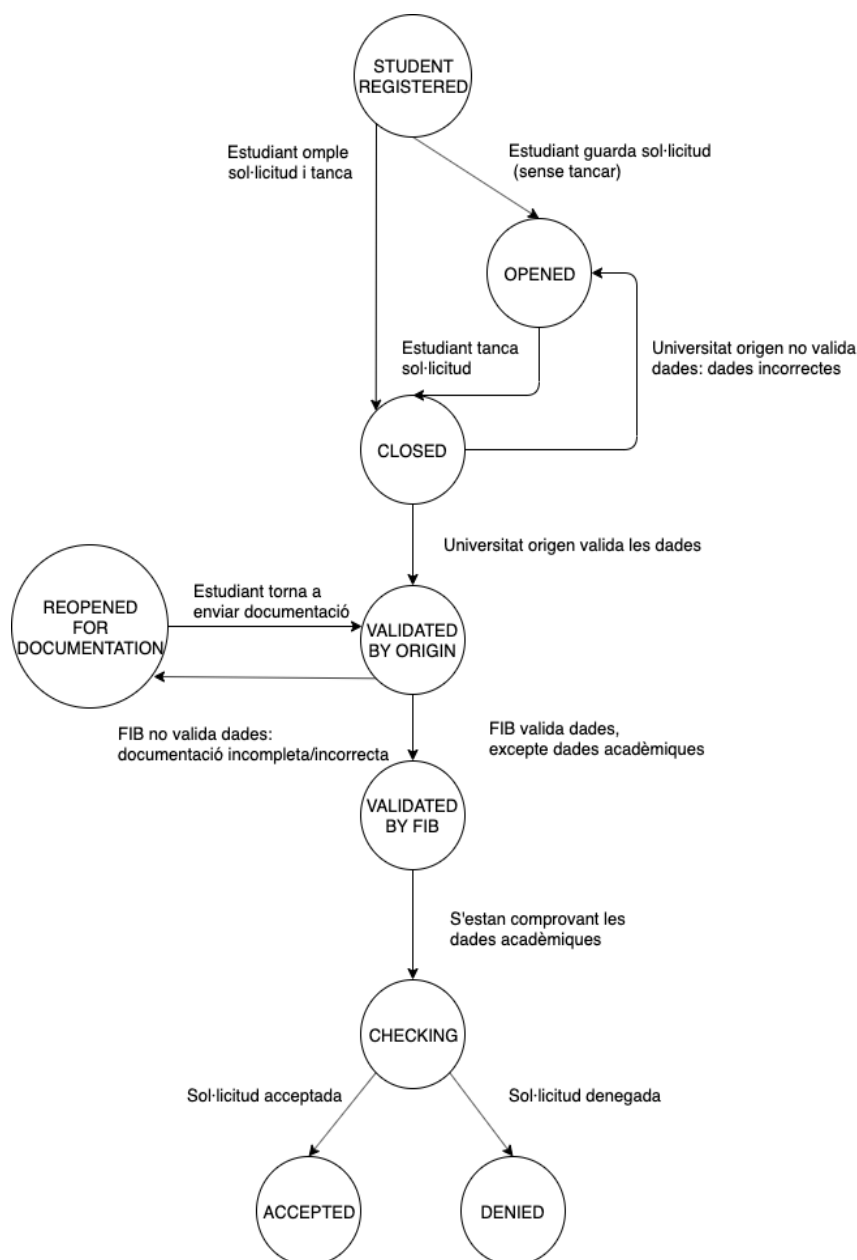
6.1.3. Diagrames d'estats

De l'anterior apartat, observem que tant la *Nominació* com la *Sol·licitud* tenen un atribut estat. A continuació podem veure els diagrames d'estats de cada una:



Imatge 12: Diagrama d'estats d'una nominació. Font: elaboració pròpia.

- **Nominated:** és el primer estat de la nominació, on ja tenim totes les dades de l'estudiant nominat.
- **Registration mail sent:** estat en el qual ja s'ha enviat el correu perquè l'estudiant es registri al sistema.

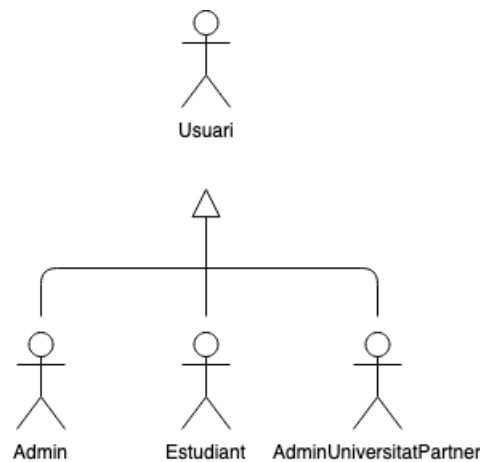


Imatge 13: Diagrama d'estats d'una sol·licitud. Font: elaboració pròpia.

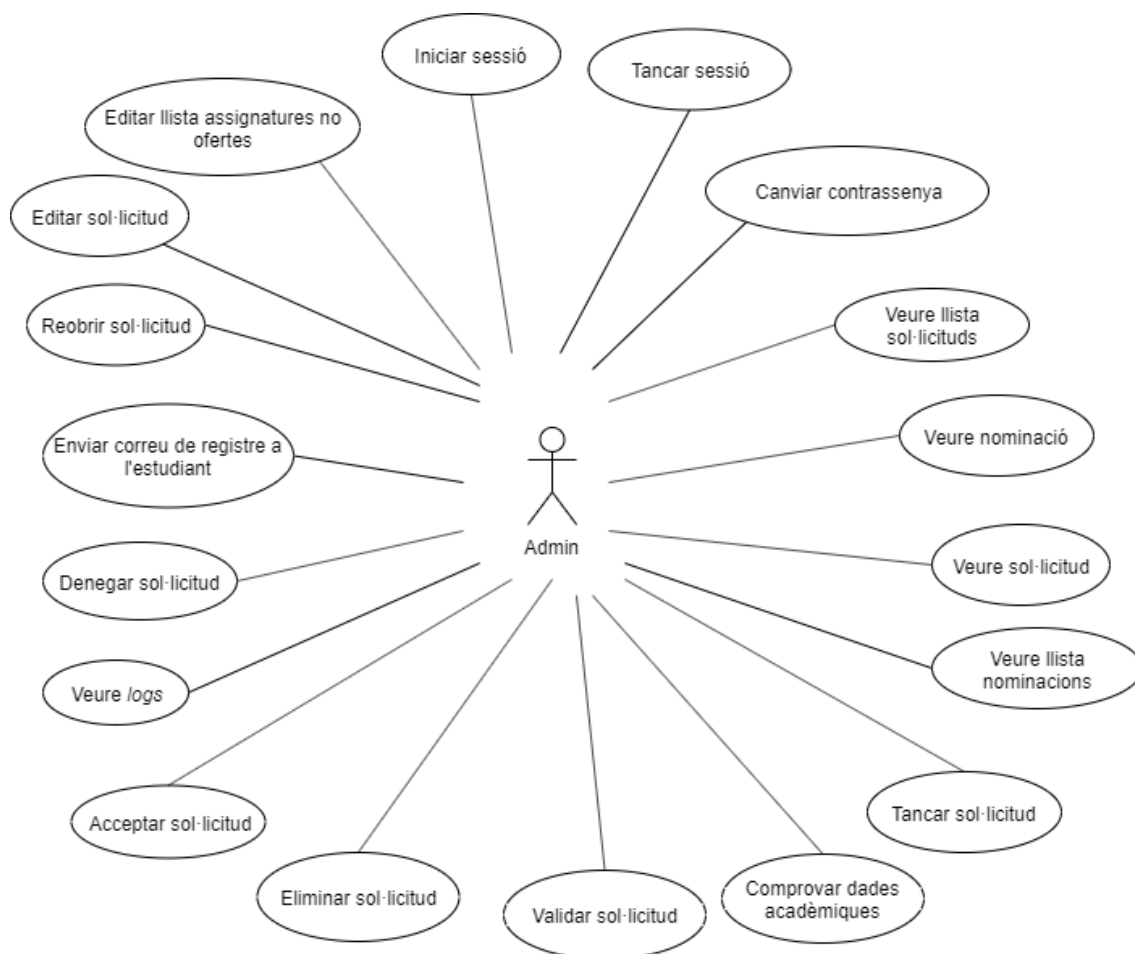
- **Student registered:** l'estudiant es registra al sistema.
- **Opened:** l'estudiant ha guardat la seva sol·licitud.
- **Closed:** l'estudiant ha acabat d'omplir les dades de la sol·licitud i vol que es revisin. Aquesta revisió la farà la universitat origen, si verifica les dades passarà a *validated by origin*. En cas contrari, obriran la sol·licitud perquè l'estudiant pugui canviar aquelles dades incorrectes.
- **Validated by origin:** la universitat origen verifica que les dades que ha omplert l'estudiant són correctes i que està d'acord amb les assignatures que ha escollit.

- **Reopened for documentation:** la FIB ha revisat les dades i indica que hi ha alguna documentació incompleta o errònia.
- **Validated by FIB:** la FIB indica que les dades i la documentació de la sol·licitud són correctes, a falta de revisar que es compleixen els requisits entre les assignatures que ha escollit i que no ha excedit el nombre de crèdits.
- **Checking:** la FIB està revisant les assignatures.
- **Accepted:** la sol·licitud ha estat acceptada.
- **Denied:** la sol·licitud ha estat denegada.

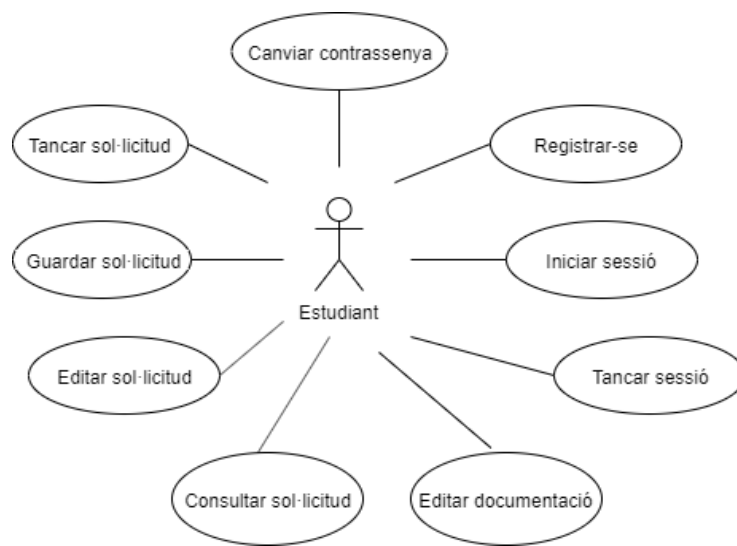
6.1.4. Diagrames de casos d'ús



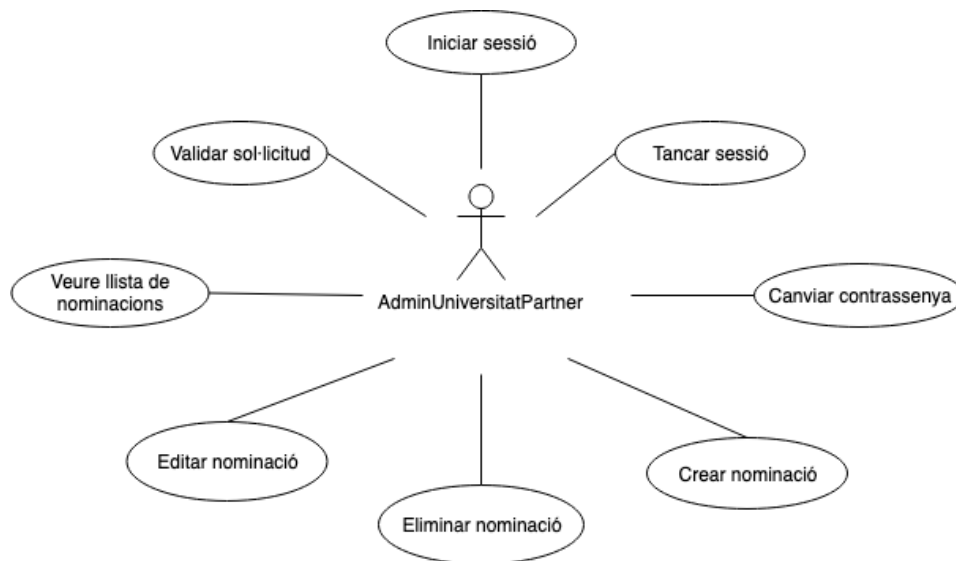
Imatge 14: Usuaris del sistema. Font: elaboració pròpia.



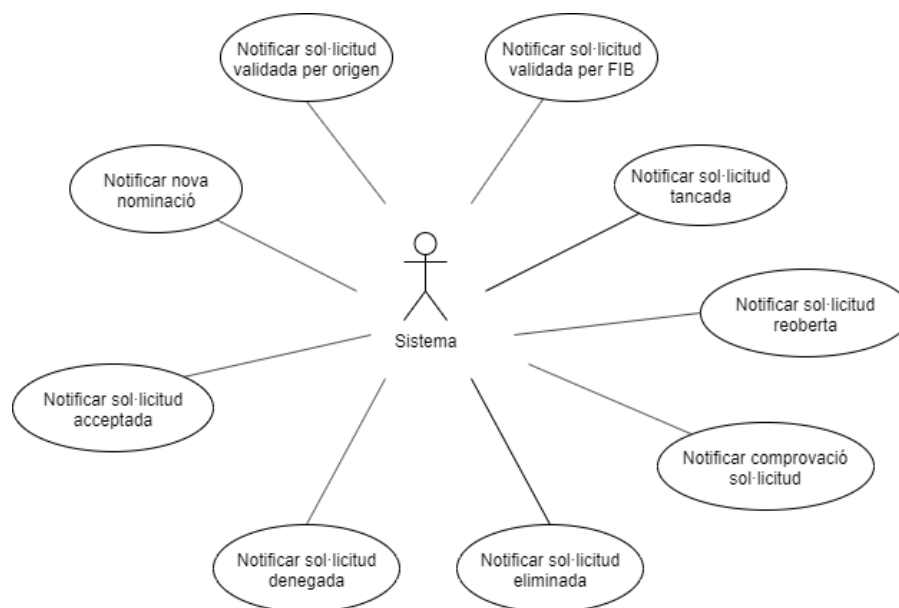
Imatge 15: Diagrama de casos d'ús de l'usuari administrador. Font: elaboració pròpia.



Imatge 16: Diagrama de casos d'ús de l'usuari estudiant. Font: elaboració pròpia.



Imatge 17: Diagrama de casos d'ús de l'usuari administrador de la universitat partner. Font: elaboració pròpia.



Imatge 18: Diagrama de casos d'ús del sistema. Font: elaboració pròpia.

6.1.5. Brief style

6.1.5.1. Administrador RRII

- **Iniciar sessió:** L'administrador de l'àrea de RRII inicia sessió al sistema.
- **Tancar sessió:** L'administrador de l'àrea de RRII tanca sessió i surt del sistema.
- **Veure llista sol·licituds:** L'administrador de l'àrea de RRII consulta la llista amb totes les sol·licituds d'estàncies acadèmiques.
- **Veure nominació:** L'administrador de l'àrea de RRII consulta les dades d'una nominació.
- **Veure sol·licitud:** L'administrador de l'àrea de RRII consulta les dades d'una sol·licitud d'estància acadèmica.
- **Veure llista nominacions:** L'administrador de l'àrea de RRII consulta la llista amb totes les nominacions.
- **Tancar sol·licitud:** L'administrador de l'àrea de RRII tanca una sol·licitud de manera que l'estudiant ja no pot modificar les dades.
- **Eliminar sol·licitud:** L'administrador de l'àrea de RRII elimina una sol·licitud.

- **Acceptar sol·licitud:** L'administrador de l'àrea de RRII accepta una sol·licitud de manera que l'estudiant és admès a la FIB.
- **Veure logs:** L'administrador de l'àrea de RRII consulta els *logs* d'una sol·licitud.
- **Denegar sol·licitud:** L'administrador de l'àrea de RRII denega una sol·licitud d'estància acadèmica.
- **Enviar correu de registre a l'estudiant:** L'administrador de l'àrea de RRII envia un correu a l'estudiant amb la informació necessària per registrar-se i accedir al sistema.
- **Reobrir una sol·licitud:** L'administrador de l'àrea de RRII torna a obrir una sol·licitud perquè l'estudiant pugui modificar la documentació.
- **Editar llista d'assignatures no obertes:** L'administrador de l'àrea de RRII edita la llista d'assignatures que no estan obertes als estudiants *incoming*.
- **Validar sol·licitud:** L'administrador de l'àrea de RRII indica que les dades i la documentació de la sol·licitud són vàlides (excepte les dades acadèmiques).
- **Comprovar dades acadèmiques:** L'administrador de l'àrea de RRII indica que s'estan comprovant les dades acadèmiques.
- **Editar sol·licitud:** L'administrador de l'àrea de RRII edita la sol·licitud d'un estudiant.

6.1.5.2. Administrador de la universitat *partner*

- **Iniciar sessió:** L'administrador de la universitat *partner* inicia sessió al sistema.
- **Tancar sessió:** L'administrador de la universitat *partner* tanca sessió del sistema.
- **Canviar contrasenya:** L'administrador de la universitat *partner* canvia la contrasenya per accedir al sistema.
- **Crear nominació:** L'administrador de la universitat *partner* crea una nominació.

- **Eliminar nominació:** L'administrador de la universitat *partner* elimina una nominació.
- **Editar nominació:** L'administrador de la universitat *partner* edita les dades d'una nominació ja existent.
- **Validar sol·licitud:** L'administrador de la universitat *partner* valida una sol·licitud d'estància acadèmica.
- **Veure llista de nominacions:** L'administrador de la universitat *partner* consulta la llista amb totes les nominacions.

6.1.5.3. Estudiant

- **Registrar-se:** L'estudiant es registra al sistema.
- **Iniciar sessió:** L'estudiant inicia sessió al sistema.
- **Tancar sessió:** L'estudiant tanca sessió del sistema.
- **Canviar contrasenya:** L'estudiant canvia la contrasenya per accedir al sistema.
- **Editar sol·licitud:** L'estudiant edita les dades de la seva sol·licitud d'estància acadèmica.
- **Consultar sol·licitud:** L'estudiant consulta les dades de la seva sol·licitud d'estància acadèmica.
- **Tancar sol·licitud:** L'estudiant tanca la seva sol·licitud de manera que ja no pot modificar les dades.
- **Editar documentació:** L'estudiant edita la documentació de la seva sol·licitud.

6.1.5.4. Sistema

- **Notificar sol·licitud validada per origen:** El sistema envia una notificació a l'administrador de l'àrea de RRII i a l'estudiant informant que la sol·licitud ha estat validada per l'administrador de la universitat de l'estudiant sol·licitant.

- **Notificar sol·licitud validada per FIB:** El sistema envia una notificació a l'estudiant informant que la seva sol·licitud ha estat validada per la FIB (excepte les dades acadèmiques).
- **Notificar sol·licitud tancada:** El sistema envia una notificació a l'administrador de la universitat de l'estudiant informant que ja pot validar la seva sol·licitud. A més, si és l'administrador de RRH qui la tanca el sistema envia també una notificació a l'estudiant per informar que ja no pot editar les dades.
- **Notificar sol·licitud reoberta:** El sistema envia una notificació a l'estudiant informant que la seva sol·licitud ha estat oberta de nou per poder editar les dades.
- **Notificar sol·licitud eliminada:** El sistema envia una notificació a l'estudiant informant que s'ha eliminat la seva sol·licitud del sistema.
- **Notificar sol·licitud denegada:** El sistema envia una notificació a l'estudiant informant que la seva sol·licitud d'estància acadèmica ha estat denegada.
- **Notificar sol·licitud acceptada:** El sistema envia una notificació a l'estudiant informant que la seva sol·licitud d'estància acadèmica ha estat acceptada, adjuntant la carta d'acceptació i el *Learning Agreement*.
- **Notificar nova nominació:** El sistema envia una notificació a l'administrador de l'àrea de RRH de la FIB informant que s'ha afegit una nominació al sistema.
- **Notificar comprovació sol·licitud:** El sistema envia una notificació a l'estudiant informant que s'estan comprovant les dades acadèmiques de la seva sol·licitud.

6.1.6. Especificació completa dels casos d'ús

En aquest apartat detallarem els casos d'ús principals més importants, aquells que aporten el *mínimum valuable product* al projecte.

6.1.6.1. Cas d'ús (CU01): Editar sol·licitud

Actor primari: Estudiant.

Disparador: L'estudiant vol editar la seva sol·licitud d'estància acadèmica.

Precondicions: L'estudiant ha estat nominat per la seva universitat i ha entrat al sistema.

Escenari principal:

1. L'estudiant selecciona l'opció d'editar la sol·licitud.
2. El sistema mostra els camps modificables.
3. L'estudiant omple o modifica les dades que vol canviar.
4. L'estudiant indica que ha acabat de modificar les dades.
5. El sistema guarda la sol·licitud.
6. El sistema mostra la sol·licitud amb les dades actualitzades.

Extensions:

- 3a. L'estudiant vol cancel·lar la modificació.
 - 3a1. L'estudiant selecciona l'opció de cancel·lar.
 - 3a2. El sistema mostra la sol·licitud com estava abans de seleccionar l'opció d'editar.
- 4a. Falta per omplir alguns dels camps.
 - 4a1. El sistema informa que alguna de les dades requerides no s'ha omplert i li comunica quin és.
 - 4a2. L'estudiant continua editant la sol·licitud al pas 3.

6.1.6.2. Cas d'ús (CU02): Crear nominació

Actor primari: Administrador universitat *partner*.

Disparador: L'administrador de la universitat *partner* vol nominar un estudiant.

Precondicions: L'administrador de la universitat *partner* ha iniciat sessió i té almenys un acord bilateral amb la FIB.

Escenari principal:

1. L'administrador de la universitat *partner* selecciona l'opció de nominar un estudiant.
2. El sistema mostra els camps que ha d'omplir.
3. L'administrador de la universitat *partner* omple els camps amb les dades de l'estudiant i de l'estància.
4. L'administrador de la universitat *partner* indica que ha acabat d'omplir les dades.
5. El sistema guarda la nominació i envia una notificació a la FIB.
6. El sistema mostra la llista de les nominacions amb la que s'acaba de crear.

Extensions:

- 3a. L'administrador de la universitat *partner* vol cancel·lar la creació de la nominació.
 - 3a1. La universitat selecciona l'opció de cancel·lar.
 - 3a2. El sistema torna a la llista de nominacions.

4a1. L'administrador de la universitat *partner* ja no pot nominar més estudiants segons el contracte bilateral.

4a1-1. El sistema mostra un missatge informant que ja no es poden nominar més estudiants.

4a2. Falta per omplir algunes dades.

4a2-1. El sistema informa que alguna de les dades requerides no s'ha omplert i li comunica quin és.

4a2-2. L'administrador de la universitat *partner* continua editant la sol·licitud al pas 3.

6.1.6.3. Cas d'ús (CU03): Editar nominació

Actor primari: Administrador universitat *partner*.

Disparador: La universitat vol editar una nominació.

Precondicions: L'administrador de la universitat *partner* ha iniciat sessió i la nominació existeix. L'estudiant nominat no s'ha registrat al sistema.

Escenari principal:

1. La universitat selecciona la nominació que vol editar.
2. La universitat selecciona l'opció d'editar.
3. El sistema mostra els camps que es poden editar.
4. La universitat modifica els camps que desitgi.
5. La universitat indica que ha acabat d'omplir les dades.
6. El sistema guarda la nominació amb les dades actualitzades.

Extensions:

4a. La universitat vol cancel·lar l'edició de la nominació.

4a1. La universitat selecciona l'opció de cancel·lar.

4a2. La universitat torna a la llista de nominacions.

5a. La universitat ha seleccionat un període on ja no es poden nominar més estudiants segons el contracte bilateral.

5a1. El sistema mostra un missatge informant que en aquell període ja no es poden nominar més estudiants.

6.1.7. Històries d'usuari

6.1.7.1. Història d'usuari (HU01): Editar sol·licitud

Com a estudiant usuari del sistema, **jo vull** poder editar la meva sol·licitud d'estància acadèmica **de manera que** la FIB pugui revisar les meves dades i decidir si sóc admès a la facultat.

Criteris d'acceptació:

- L'estudiant ha de poder editar la seva sol·licitud sempre que estigui oberta.
- L'estudiant pot cancel·lar l'edició en qualsevol moment.
- El sistema ha de mostrar un missatge d'error si no omple totes les dades requerides.

Escenari 1: Editar una sol·licitud

Donat un estudiant nominat loguejat al sistema que vol editar la seva sol·licitud

Quan omple o modifica les dades necessàries

Aleshores el sistema guarda la sol·licitud amb les noves dades

Escenari 2: Cancel·lar la modificació

Donat un estudiant nominat loguejat al sistema que vol cancel·lar la modificació de les dades de la seva sol·licitud

Quan decideix cancel·lar la modificació de la seva sol·licitud

Aleshores el sistema torna a la pantalla anterior sense modificar cap dada

Escenari 3: No s'han omplert totes les dades requerides

Donat un estudiant nominat loguejat al sistema que vol editar la seva sol·licitud

Quan no omple totes les dades requerides

Aleshores el sistema mostra un missatge d'error indicant que falten algunes dades per omplir

6.1.7.2. Història d'usuari (HU02): Crear nominació

Com a administrador d'una universitat *partner* loguejat al sistema, **jo vull** poder crear una nominació **de manera que** la FIB tingui coneixement que hi ha un estudiant que vol realitzar una estància acadèmica.

Criteris d'acceptació:

- L'administrador de la universitat *partner* podrà nominar en tot moment dins del termini de nominacions establert per la facultat.

- El sistema mostrarà un missatge d'error quan, segons el conveni bilateral, ja no es puguin nominar més estudiants en el període que s'ha escollit.
- El sistema mostrarà un missatge d'error quan l'administrador de la universitat *partner* no ompli totes les dades requerides al nominar un estudiant.

Escenari 1: Crear nominació

Donat un administrador de la universitat *partner* loguejat al sistema

Quan omple les dades necessàries per crear una nominació

Aleshores el sistema guarda les dades de la nova nominació

Escenari 2: Cancel·lar nominació

Donat un administrador de la universitat *partner* loguejat al sistema

Quan decideix cancel·lar la creació de la nominació

Aleshores el sistema torna a la pantalla anterior i no guarda cap dada

Escenari 3: No es poden nominar més estudiants pel període seleccionat

Donat un administrador de la universitat *partner* loguejat al sistema

Quan omple les dades per crear una nominació i selecciona un període on ja no es poden nominar més estudiants

Aleshores el sistema mostra un missatge d'error informant que, d'acord amb el contracte bilateral, ja no es poden nominar més estudiants en el període seleccionat

Escenari 4: No s'han omplert totes les dades requerides

Donat un administrador de la universitat *partner* loguejat al sistema

Quan no omple totes les dades requerides per crear una nominació

Aleshores el sistema mostra un missatge d'error informant que no s'han omplert totes les dades requerides per crear una nominació

6.1.7.3. Història d'usuari (HU03): Editar nominació

Com a administrador d'una universitat *partner* loguejat al sistema, **jo vull** poder editar una nominació **de manera que** les dades estiguin correctes.

Criteris d'acceptació

- L'administrador de la universitat *partner* podrà editar una nominació sempre que l'estudiant nominat no s'hagi registrat al sistema.
- El sistema mostrarà un missatge d'error quan, segons el conveni bilateral, ja no es puguin nominar més estudiants en el període que s'ha escollit.

Escenari 1: Editar nominació

Donat un administrador de la universitat *partner* loguejat al sistema

Quan modifica les dades d'una nominació ja existent

Aleshores el sistema guarda les dades actualitzades de la nominació

Escenari 2: Cancel·lar la modificació de la nominació

Donat un administrador de la universitat *partner* loguejat al sistema

Quan decideix cancel·lar la modificació de la nominació

Aleshores el sistema no guarda cap dada i torna a la pantalla anterior

Escenari 3: No es poden nominar més estudiants pel període seleccionat

Donat un administrador de la universitat *partner* loguejat al sistema

Quan modifica les dades d'una nominació i selecciona un període on ja no es poden nominar més estudiants

Aleshores el sistema mostra un missatge d'error informant que, d'acord amb el contracte bilateral, ja no es poden nominar més estudiants en el període seleccionat

6.2. Requisits no funcionals

En aquest apartat descriurem els requisits no funcionals, que són aquells que indiquen quines qualitats ha de tenir el sistema, fent servir la plantilla de requisits Volere [\[16\]](#).

A continuació es mostra una taula per cada un d'ells, als quals s'ha assignat una prioritat en una escala de l'1 al 5.

6.2.1. Aparença

Número	1
Nom i tipus segons Volere	Aparença (10a)
Descripció i justificació	L'aparença de l'aplicació ha de seguir els patrons de la resta d'aplicacions de la FIB, per tal de no perdre la consistència entre aquestes ja que es tracta d'una aplicació oficial.
Criteri de satisfacció	Es farà servir la plantilla web de l'API de la FIB.
Prioritat	3

6.2.2. Facilitat d'ús

Número	2
Nom i tipus segons Volere	Facilitat d'ús (11a)
Descripció i justificació	El sistema ha de ser senzill d'utilitzar pels usuaris i el seu funcionament ha de ser similar i familiar a l'anterior aplicació. D'aquesta manera, volem aconseguir que els usuaris cometin el nombre mínim d'errors possibles.
Criteri de satisfacció	Es preguntarà a un conjunt d'usuaris i el 90% d'aquests ha de considerar que és fàcil d'utilitzar.
Prioritat	4

6.2.3. Personalització i internacionalització

Número	3
Nom i tipus segons Volere	Personalització i internacionalització (11b)
Descripció i justificació	L'idioma de l'aplicació serà l'anglès ja que està destinada a un públic internacional.
Criteri de satisfacció	Tot el sistema serà en anglès.
Prioritat	3

6.2.4. Aprenentatge

Número	4
Nom i tipus segons Volere	Aprenentatge (11c)
Descripció i justificació	El sistema no requerirà de cap formació prèvia per la seva utilització.
Criteri de satisfacció	Es preguntarà a un conjunt d'usuaris i el 90% d'aquests ha de considerar no ha necessitat cap formació prèvia per utilitzar l'aplicació.
Prioritat	3

6.2.5. Fiabilitat i disponibilitat

Número	5
Nom i tipus segons Volere	Fiabilitat i disponibilitat (12d)
Descripció i justificació	L'aplicació ha d'estar disponible durant els períodes d'inscripció dels estudiants <i>incoming</i> i, durant aquest temps, ha de funcionar sense errors durant el màxim de temps possible.
Criteri de satisfacció	El <i>hosting</i> es farà als servidors de la FIB.
Prioritat	5

6.2.6. Interacció amb sistemes adjacents

Número	6
Nom i tipus segons Volere	Interacció amb sistemes adjacents (13c)
Descripció i justificació	L'aplicació ha d'utilitzar la mateixa base de dades que fa servir l'aplicació de gestió de dades dels estudiants de mobilitat internacional ja que necessita consultar algunes de les dades que emmagatzema.
Criteri de satisfacció	Auto-explicatiu.
Prioritat	5

6.2.7. Implantació

Número	7
Nom i tipus segons Volere	Implantació (13d)
Descripció i justificació	L'aplicació no requerirà cap instal·lació ja que es tracta d'una aplicació web.
Criteri de satisfacció	El sistema ha de ser accessible des de qualsevol navegador de les últimes versions estables, sense cap instal·lació prèvia.
Prioritat	5

6.2.8. Accés

Número	8
Nom i tipus segons Volere	Accés (15a)
Descripció i justificació	Podran accedir a l'aplicació web únicament les universitats <i>partner</i> , els estudiants que hagin estat nominats per aquestes i el personal de l'àrea de RRII de la FIB.
Criteri de satisfacció	Auto-explicatiu.
Prioritat	5

6.2.9. Privacitat

Número	9
Nom i tipus segons Volere	Privacitat (15c)
Descripció i justificació	L'aplicació web garantirà la confidencialitat de les dades introduïdes ja que aquestes són de caràcter sensible.
Criteri de satisfacció	S'informarà als usuaris que les dades que introdueixin als formularis només seran tractades amb finalitat purament acadèmica. Es seguirà el Reglament (UE) 2016/679 del Parlament i del Consell, de 27 d'abril de 2016, "relatiu a la protecció de les persones físiques pel que fa al tractament de dades personals i a la lliure circulació d'aquestes dades" [17]
Prioritat	5

7. Disseny i implementació

7.1. Tecnologies utilitzades

7.1.1. Django

Django [\[18\]](#) és un *framework* de codi obert escrit en Python [\[19\]](#), llenguatge interpretat de codi obert que suporta orientació a objectes, que permet crear aplicacions web. La primera versió es va llançar l'any 2005 i la última versió estable és la 2.2 llançada l'abril de 2019, per la qual cosa veiem que és un *framework* en desenvolupament constant. Les principals característiques de Django són:

- Utilitza el patró arquitectònic Model-Vista-Controlador (MVC).
- Controladors genèrics que contenen els mètodes HTTP bàsics.
- Herència de plantilles, amb la qual cosa estalvies haver de repetir innecessàriament codi.
- Un despatxador de URLs amb expressions regulars.
- Mapejador objecte-relacional (ORM, de l'anglès *Object-Relational Mapping*), que permet que les dades dels objectes creats a la nostra aplicació siguin gravades a la base de dades independentment de la tecnologia d'aquesta.

Així doncs, el fet d'utilitzar aquest *framework* comportarà que també s'utilitzin els llenguatges HTML, JavaScript i CSS per la implementació de les plantilles.

7.1.2. Bootstrap

Bootstrap [\[20\]](#) és una llibreria de codi obert de *front-end* que ofereix diverses plantilles per millorar l'aparença de pàgines i aplicacions web i a més, ens proporciona eines per crear pàgines *responsive*. En el nostre cas hem adaptat el tema personalitzat de Bootstrap que es fa servir a l'API de la FIB per utilitzar-lo també a l'aplicació web i així tenir una consistència en l'aparença de les aplicacions de la FIB.

7.1.3. jQuery

jQuery [\[21\]](#) és una llibreria de *software* lliure de JavaScript que simplifica el desenvolupament en aquest llenguatge comprimint diverses línies de codi en un sol mètode. A més, ofereix un mètode

per fer peticions AJAX, que ens serveix per fer les crides a l'API de la FIB i obtenir les dades que necessitem.

7.1.4. Oracle Database

Per a la base de dades s'ha escollit Oracle [\[22\]](#), un sistema molt conegut de gestor de base de dades de llicència privada orientada a objectes. És la base de dades que ja s'utilitzava per emmagatzemar informació sobre les universitats, els convenis bilaterals, i guardar les dades dels estudiants un cop són acceptats.

7.1.5. PyCharm

PyCharm [\[23\]](#) és un entorn de desenvolupament integrat (IDE, de l'anglès *Integrated Development Environment*) desenvolupat per JetBrains, que s'utilitza per desenvolupar aplicacions en Python principalment. Nosaltres utilitzem la versió Professional, ja que ens ofereix una integració amb un servidor local per poder desenvolupar l'aplicació en un entorn local. Altres característiques:

- Suport pel *framework* Django
- Depurador de Python
- Integració amb Git
- Integració amb testos unitaris

7.1.6. DBeaver

DBeaver [\[24\]](#) és una eina lliure i oberta d'administració de base de dades i client SQL amb suport per múltiples sistemes gestors de base de dades, entre ells Oracle.

7.1.7. Git

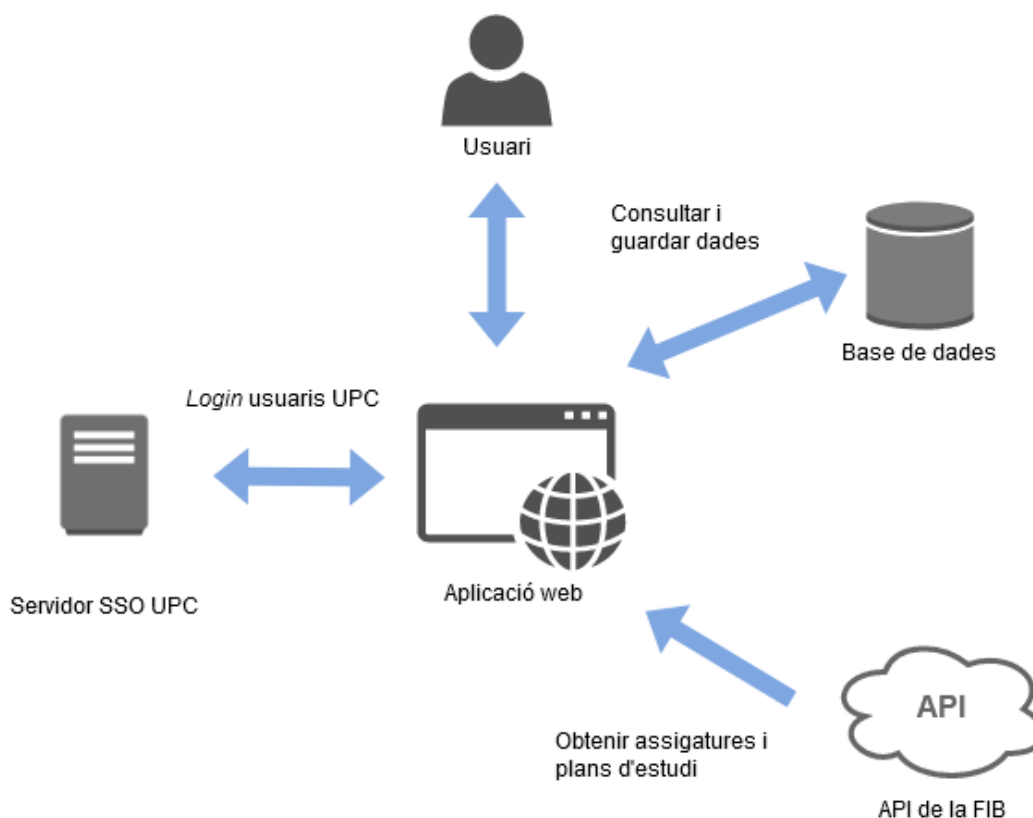
Git és una eina de *software* lliure que serveix per realitzar un control de versions del nostre codi, amb la qual podem veure l'historial complet de canvis realitzats en el codi des de la creació de l'aplicació. També ofereix un sistema de branques, que ens serveix per desenvolupar noves funcionalitats en una altra branca sense tocar cap part de la versió estable l'aplicació que s'emmagatzema a una branca principal.

7.1.8. GitLab UPC

El GitLab de la UPC [\[25\]](#) és un servei web de codi obert basat en GitLab [\[26\]](#) i, a la vegada, basat en Git (d'aquí el seu nom) que serveix per gestionar repositoris. En aquesta web emmagatzemem el codi font de la nostra aplicació, així com altres codis fonts d'altres aplicacions de la FIB gestionades per l'inLab.

7.2. Arquitectura del projecte

A continuació es mostra un esquema de l'arquitectura del projecte:



Imatge 19: Esquema de l'arquitectura del projecte. Font: elaboració pròpia.

Com es pot veure, l'aplicació web es connecta a la base de dades per emmagatzemar dades pròpiament de l'aplicació i consultar informació sobre les universitats i els convenis, entre d'altres.

A més, es fan peticions a l'API de la FIB per obtenir informació sobre les assignatures de la FIB i sobre els plans d'estudis oberts. Això últim és una novetat a l'aplicació, ja que anteriorment aquesta informació es consultava a la base de dades, la qual cosa resulta un accés innecessari a la base de

dades ja que aquestes dades les podem obtenir de l'API de la FIB sense haver de crear cap taula ni emmagatzemar cap informació, simplement realitzant una petició a les URLs on es trobin aquests recursos.

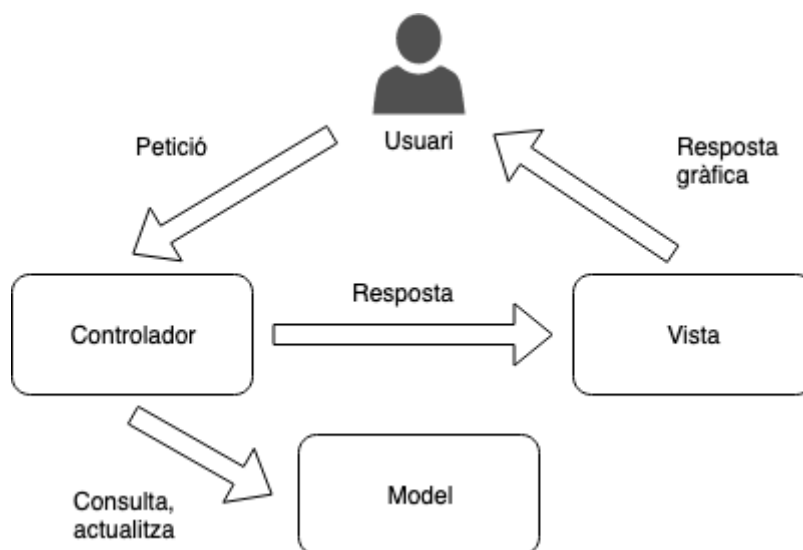
Pel que fa a l'autenticació d'usuaris, s'utilitza l'SSO de la UPC; concretament, pels usuaris administradors de l'Àrea de Relacions Internacionals. Amb aquest mètode aquestes persones no hauran de registrar-se al sistema per loguejar-se, sinó que iniciant sessió amb les credencials que fan servir als sistemes web de la UPC podran accedir a la nostra aplicació web.

7.3. Model-Vista-Controlador (MVC)

Com s'ha comentat anteriorment l'aplicació web utilitzarà el patró arquitectònic Model-Vista-Controlador (MVC, de l'anglès *Model-View-Controller*), que consisteix en tres components:

- **Model:** Conté la representació de les dades i s'encarrega de gestionar-les i actualitzar-les accedint a la base de dades. És independent de la tecnologia de la base de dades, és a dir, a l'hora de programar els models no ens hem de preocupar per si la base de dades és Oracle o SQLite, per exemple. En el cas de Django, és el *framework* qui s'encarrega d'adaptar les sentències SQL al sistema gestor de base de dades que s'utilitza.
- **Vista:** Mostra la informació en un format llegible per l'usuari final. Parlem doncs, de les plantilles HTML o de la interfícies d'usuari en el cas d'una aplicació d'escriptori.
- **Controlador:** Reb les peticions de l'usuari i les processa: sol·licita les dades necessàries al model, les modifica segons el que hagi demanat l'usuari i les entrega a la vista perquè les mostri adequadament.

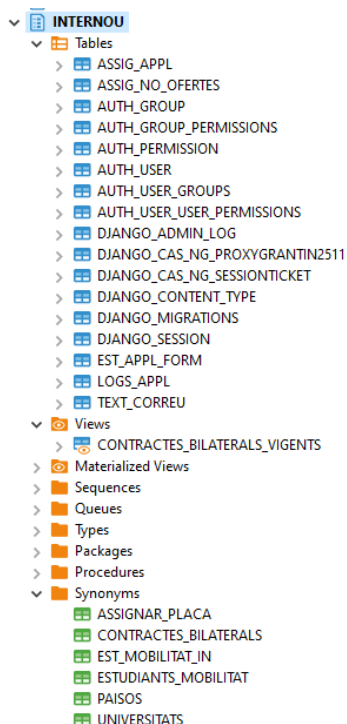
En el següent esquema podem veure el comportament d'aquests 3 components:



Imatge 20: Esquema del patró arquitectònic MVC. Font: elaboració pròpia.

7.4. Estructura de la base de dades

Com ja s'ha comentat en anteriors apartats, a la base de dades tindrem tant dades pròpies de l'aplicació com dades que es fan servir per l'aplicació de gestió de dades de l'Àrea de Relacions Internacionals. En la següent imatge es mostra l'estructura de la base de dades:



Imatge 21: Taules de la base de dades. Font: elaboració pròpia.

Podem veure clarament que les taules que comencen per *auth_* i *django_* són taules gestionades per Django. Les més importants són:

- **auth_user_group:** És on es guarden els grups als que pertany cada usuari. A la nostra aplicació tenim dos tipus de grups per diferenciar els usuaris: el grup estudiants i un grup per cada universitat, el nom del qual és l'identificador de la mateixa. D'aquesta manera, quan un estudiant es registri pertanyerà al grup d'estudiants i els usuaris administradors d'una universitat *partner* pertanyeran al grup que tindrà com a nom el codi de la institució. S'ha decidit fer-ho d'aquesta manera ja que considerem que és una forma senzilla d'obtenir quin tipus d'usuari és cada usuari registrat al sistema.
- **auth_user_permissions:** En aquesta taula, com el seu nom indica, es guarden els permisos que té cada usuari. Segons aquests permisos, distingirem dos tipus d'usuaris: els usuaris comuns i els administradors o superusuaris. Aquests últims seran els usuaris de l'Àrea de RRII, ja que podran accedir a la pantalla de gestió de l'aplicació, des de la qual podran veure tots els usuaris registrats, els grups als quals pertanyen i més informació delicada.
- **auth_user:** Taula on es guarda la informació dels usuaris tant per iniciar sessió (*username* i *password* encriptat) com informació extra (correu, nom, cognoms...).
- **django_migrations:** Les *migrations* són un sistema de control de versions de la base de dades. Cada cop que es fa un canvi a un model, Django es guarda a aquesta taula la data dels canvis i els propaga a la taula corresponent al model actualitzat.
- **països:** Taula on es guarden els noms dels països en català, castellà i anglès. Aquesta taula la reaprofitem de l'anterior aplicació i també es fa servir a l'aplicació d'escriptori de gestió de dades.
- **universitats:** Aquí es guarda informació sobre la universitat com el seu identificador, el nom de la institució, el país on es troba, etc. Aquesta taula també es fa servir a l'aplicació d'escriptori de gestió de dades.
- **text_correu:** Conté les plantilles dels correus de notificació que s'envien cada cop que una nominació o sol·licitud canvia d'estat. Aquesta taula la reaprofitem de l'anterior aplicació.
- **logs_appl:** Conté els *logs* que es van creant a mesura que es fan accions sobre una nominació o sol·licitud. Per exemple, es guarda la data de nominació i dels últims canvis a una sol·licitud.

- **assig_appl:** Guarda les assignatures escollides a una sol·licitud. Concretament, es guarda l'identificador únic per cada assignatura, l'identificador de la sol·licitud i el quadrimestre en que es vol fer l'assignatura.
- **est_appl_form:** En aquesta taula s'emmagatzema tota la informació sobre les nominacions i sol·licituds. Quan una universitat nomina un estudiant, s'afegeix un nou registre a la taula amb un identificador, les dades de l'estudiant nominat i les dades de l'estància acadèmica. Quan l'estudiant es registra, s'afegeix el seu nom d'usuari a la columna *username* d'aquest registre, és a dir, vinculem l'usuari amb aquesta nominació. Més endavant, quan l'estudiant omple i guarda les dades de la seva sol·licitud s'actualitza aquesta fila amb les noves dades.

S'ha decidit aquest disseny per dues raons principals. La primera, perquè una sol·licitud conté les dades de la nominació de manera que no caldria separar nominació i sol·licitud en dues taules. La segona, perquè els estudiants es registren després de ser nominats de manera que quan entren per primer cop al sistema ja tenim les seves dades personals, per la qual cosa si guardéssim la seva informació personal en una taula a part estariem duplicant informació.

- **contractes_bilaterals:** Conté les dades sobre tots els contractes bilaterals, vigents i no vigents. Concretament, la universitat amb la qual tenim contracte, el màxim de quadrimestres que poden nominar, l'any de caducitat del contracte i el nom del programa de mobilitat. També es fa servir a l'aplicació d'escriptori de gestió de dades
- **contractes_bilaterals_vigents:** És una vista sobre la taula *contractes_bilaterals* seleccionant aquells contractes que estan vigents.
- **est_mobilitat:** Aquí es guarden les dades personals de tots els estudiants un cop són acceptats a un programa de mobilitat, tant estudiants *incoming* com estudiants *outgoing*. Concretament, el seu carnet d'identitat (DNI o passaport), el nom i primer cognom i el correu electrònic. També es fa servir a l'aplicació d'escriptori.
- **est_mobilitat_in:** Taula on copiem les dades dels estudiants *incoming* un cop són acceptats, concretament el seu carnet d'identitat (DNI o passaport), telèfon i l'adreça al país d'origen. Aquesta taula es fa servir també a l'aplicació d'escriptori.

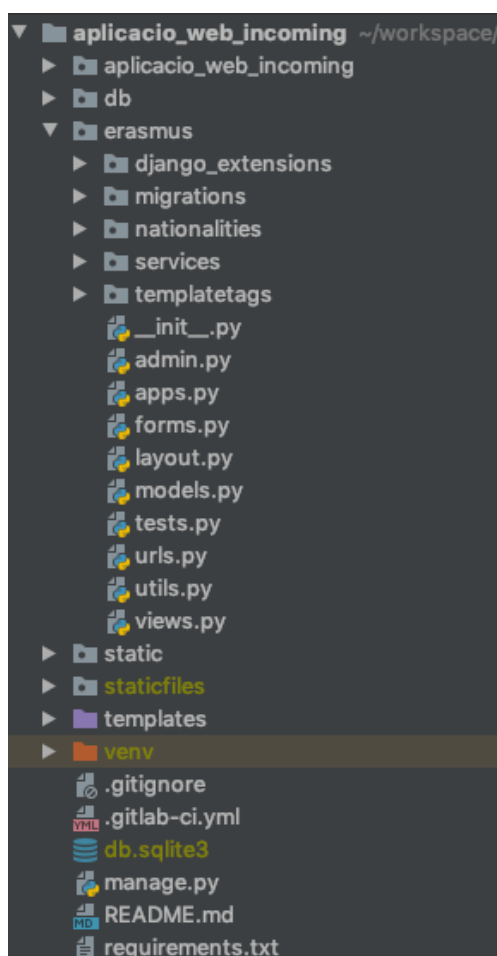
- **assignar_placa:** En aquesta taula es guarda l'estat de l'assignació de places de mobilitat, tant pels estudiants *incoming* com pels *outgoing*. Aquesta taula es fa servir també a l'aplicació d'escriptori.

7.5. Detalls de la implementació

En aquest apartat comentarem alguns detalls importants i destacats sobre la implementació del projecte.

7.5.1. Estructura del directori

A la següent imatge podem veure l'estructura del directori de l'aplicació:



Imatge 22: Estructura del directori de l'aplicació. Font: elaboració pròpia.

A continuació explicarem els directoris més rellevants:

- **aplicacio_web_incoming:** És on es troba tota la configuració de l'aplicació, com per exemple els paràmetres per connectar-se a la base de dades, el servidor de correu, la caché, els paràmetres per connectar-se al SSO UPC, etc.
- **db:** En aquest directori guardem les sentències SQL per crear i afegir dades a les taules. Això només s'ha utilitzat per treballar des de casa, ja que per seguretat no hi teníem accés a la base de dades de la FIB.
- **erasmus:** Aquí és on Django guarda les *migrations* i on nosaltres implementem les funcionalitats de l'aplicació.
 - **services:** Conté la implementació de la comunicació amb serveis externs, concretament de l'API. La comunicació amb el SSO UPC no ha calgut implementar-la ja que hem utilitzat una llibreria de Python que ens proporciona aquesta tasca.
 - **models:** Com el seu nom indica, conté la implementació de tots els models.
 - **tests:** Conté els tests unitaris.
 - **urls:** Aquí configurem els mapejos de les URLs cap als controladors.
 - **views:** Al contrari del que suggereix el seu nom, les *views* de Django són els controladors del patró MVC.
- **static:** Directori on es troben els CSS i les imatges que fa servir l'aplicació.
- **templates:** Són les vistes de l'aplicació (el component *View* del patró MVC), és a dir, les plantilles HTML.

7.5.2. Models

Com s'ha comentat a l'apartat [7.3](#), el Model és el component del patró MVC que representa una entitat de la base de dades. A la següent imatge podem veure la implementació d'un model:

```
class ContracteBilateral(models.Model):
    codi_universitat = models.CharField(max_length=20, db_column='CODI', primary_key=True)
    tipus = models.CharField(max_length=20)
    any_acad = models.IntegerField()
    quadris_in = models.IntegerField()

    class Meta:
        managed = False
        db_table = 'CONTRACTES_BILATERALS_VIGENTS'
```

Imatge 23: Un exemple de model de Django. Font: elaboració pròpia.

Observem que el codi és bastant intuïtiu. Es tracta de declarar cada columna de la taula que es vulgui utilitzar indicant quin tipus de dades conté. L'atribut *managed* de la meta-classe ens indica si aquest model representa una taula ja existent a la base de dades i, per tant, gestionada per nosaltres o, si pel contrari, volem que Django l'administri i actualitzi la taula cada cop que fem canvis en el model. En el nostre cas no volem que el gestioni Django ja que es tracta d'una taula pròpia, per tant li assignem el valor *False*.

7.5.3. Controladors

A Django els controladors s'anomenen *Views*. Cada vista és invocada quan l'usuari accedeix a la URL mapejada per aquesta, i consisteix en un mètode que retorna una resposta HTTP.

```
@login_required
@staff_member_required
def incoming_students_list(request):
    msg = ''
    agreements = []
    ordering = request.GET.get('ordering', 'last_name')
    acad_course = request.GET.get('academic_course', 'all')
    if acad_course == 'all':
        applications = Application.objects.all().order_by(ordering)
    else:
        applications = Application.objects.filter(academic_course=acad_course).order_by(ordering)

    year = datetime.datetime.now().year
    return render(request, 'applications_list.html',
        {'applications': applications, 'msg': msg,
         'agreements': agreements,
         'acad_courses': [c[0] for c in forms.get_academic_courses(start_year=year - 1, end_year=year + 4)],
         'nomination_status': NOMINATION_STATUS,
         'editable_app_status': EDITABLE_APP_STATUS,
        })
```

Imatge 24: Exemple de controlador a Django. Font: elaboració pròpia.

Veiem aquest exemple senzill d'un controlador a Django. En aquest cas, el que fa és mostrar tota la llista de sol·licituds dels estudiants, ordenada i/o filtrada pel paràmetre que hagi seleccionat l'usuari. Per fer això, realitza la consulta al model (*Application*, a l'exemple) i, un cop obtingudes les dades, les estructura en un diccionari perquè posteriorment la vista HTML les llegeixi i mostri adequadament.

7.5.4. Vistes

```
<div class="card-body">
  <ul class="list-group list-group-flush">
    <li class="list-group-item"><b>Institution:</b> {{ a.institution.nom }}</li>
    <li class="list-group-item"><b>Name:</b> {{ a.name }}</li>
    <li class="list-group-item"><b>Last name:</b> {{ a.last_name }}</li>
    <li class="list-group-item"><b>Second last name:</b> {{ a.second_last_name }}</li>
    <li class="list-group-item"><b>Email:</b> {{ a.email }}</li>
    <li class="list-group-item"><b>Current studies:</b> {{ a.current_studies }}</li>
    <li class="list-group-item"><b>Period:</b> {{ a.academic_course }} - {{ a.get_period_display }}
    </li>
    <li class="list-group-item"><b>Status:</b>
      <span class="badge
        {% if a.status == 'N' %} badge-primary
        {% elif a.status == 'S' %} badge-light
        {% elif a.status == 'R' %} badge-secondary
        {% elif a.status == 'O' %} badge-info
        {% elif a.status == 'RO' %} badge-info
        {% elif a.status == 'CL' %} badge-dark
        {% elif a.status == 'VO' %} badge-light
        {% elif a.status == 'VF' %} badge-light
        {% elif a.status == 'CH' %} badge-warning
        {% elif a.status == 'A' %} badge-success
        {% elif a.status == 'D' %} badge-danger
        {% endif %}">
        {{ a.get_status_display }}</span>
      </li>
    </ul>
  </div>
```

Imatge 25: Exemple de vista a Django. Font: elaboració pròpia.

Anomenem vistes a les plantilles HTML, que donen una resposta gràfica a les peticions dels usuaris. En el patró MVC, aquestes vistes reben les dades del controlador per ser manipulades de manera que es puguin mostrar als usuaris.

En aquest fragment de plantilla, veiem que es mostren algunes dades d'una sol·licitud mitjançant el llenguatge de plantilles proporcionat per Django anomenat Jinja2 [\[27\]](#), que també ens permet escriure sentències condicionals o bucles a la pròpia plantilla HTML.

7.5.5. Comunicació amb l'API

Per obtenir la informació sobre les assignatures de la FIB i els plans d'estudi que s'ofereixen, hem consultat l'API de la FIB.

```
def get_degrees_en():
    """
    Obtenim la descripció en anglès sobre els plans d'estudi actualment oferts pels incoming ['GRAU', 'MAI', 'MEI', 'MIRI']
    :return: Un diccionari que té com a claus els codis dels plans i com a valors la descripció dels mateixos.
    """
    resp = cache.get('degrees_en')
    if resp is None:
        http = urllib3.PoolManager()
        r = http.request('GET', settings.API_URL + '/plans_estudi?format=json&client_id=' + settings.API_CLIENT_ID,
                        headers={'Accept-Language': 'en-US'})
        data = json.loads(r.data.decode('UTF-8')).get('results')
        resp = {}
        for d in data:
            if d.get('abreviatura') in INCOMING_DEGREES:
                resp[d.get('abreviatura')] = d.get('descripcio')
        cache.set('degrees_en', resp)
    return resp
```

Imatge 26: Obtenció dels plans d'estudi a través de l'API de la FIB. Font: elaboració pròpia.

En aquesta imatge podem veure com s'obtenen les dades de l'API. Es tracta de fer una petició GET d'HTTP a la URL on es troben les dades que volem consultar, en el nostre cas els plans d'estudi. Observem que proporcionem dos paràmetres:

- **format:** El format en el qual volem rebre les dades, en aquest cas JSON.
- **client_id:** Identificador alfanumèric de la nostra aplicació perquè l'API ens reconegui com a aplicació registrada i ens retorni la resposta.

Un cop obtenim la informació desitjada adaptem l'estructura desitjada i la guardem a la caché configurada en un directori local, per tal d'estalviar-nos crides a l'API.

7.5.6. Login amb l'SSO UPC

L'SSO UPC permet que, amb un sol cop que s'hagi iniciat sessió amb les credencials de la UPC, puguem accedir a tots els sistemes que l'implementin, com ara el Racó de la FIB, l'API de la FIB i també la nostra aplicació, si es tracta d'un usuari administrador. Per aconseguir-ho, hem utilitzat la llibreria de Python *django-cas-ng* [28] que implementa tota la comunicació amb el servidor SSO amb el que desitgem realitzar l'autenticació.

A l'arxiu de configuració de Django (*settings.py*) hem afegit les següents variables:

```
# CAS Server
# Installed app: https://pypi.org/project/django-cas-ng/

CAS_SERVER_URL = "https://sso.pre.upc.edu/CAS/"

CAS_LOGOUT_COMPLETELY = True

CAS_REDIRECT_URL = 'home'
```

Imatge 27: Configuració de l'SSO. Font: elaboració pròpia.

Podem veure tres variables:

- **CAS_SERVER_URL:** La URL del servidor SSO de la UPC.
- **CAS_LOGOUT_COMPLETELY:** Si el seu valor és *True*, significa que quan l'usuari faci *logout* de l'aplicació també tancarà la sessió de tots els llocs web on hagi iniciat sessió amb el CAS.
- **CAS_REDIRECT_URL:** URL on es redireccionarà l'usuari un cop hagi iniciat sessió.

Un cop tenim el CAS configurat, només queda mapejar els controladors (*views*) que ens ofereix la llibreria, que contenen tota la comunicació amb el servidor SSO desitjat. Així doncs, a l'arxiu de configuració de les URLs, *urls.py*, hem afegit aquestes dues URLs:

```
path('cas/login/', cas_views.LoginView.as_view(), name='cas-login'),
path('cas/logout/', cas_views.LogoutView.as_view(), name='cas-logout'),
```

Imatge 28: Mapeig de les URLs de l'SSO. Font: elaboració pròpia.

7.5.7. Registre d'usuaris

El nostre sistema està destinat a un grup d'usuaris específic. Donada la sensibilitat de la informació que s'hi introdueix, no podem permetre l'accés a l'aplicació a qualsevol persona, per la qual cosa hem hagut de limitar el registre d'usuaris.

En quant a les universitats *partner*, serem nosaltres els que els donarem d'alta mitjançant l'eina d'administració que ens proporciona Django, dintre de la qual podem crear, modificar o eliminar usuaris i rols als que pertanyen aquests usuaris. Com que per una universitat por haver-hi més d'una

persona que s'encarregui de la gestió dels estudiants de mobilitat, hem decidit crear un rol per cada institució, de manera que cada usuari que pertanyi a aquesta tingui el mateix rol, i així poder gestionar els usuaris de la mateixa universitat consultant el seu valor, que correspondrà al codi de la universitat.

En quant als estudiants, només es podran registrar si reben el correu de registre, ja que l'enllaç que els proporcionem conté un *token* generat per l'aplicació que és únic i que caduca un cop l'estudiant es registra. D'aquesta manera, per accedir a la pàgina de registre de l'aplicació es necessita aquest *token*.

7.6. Proves

Per comprovar el funcionament del sistema, s'han implementat proves unitàries que han ajudat a comprovar i alhora a entendre alguns requisits que no es poden especificar al 100% i que no es poden provar manualment de manera fàcil.

Parlem concretament de les restriccions de nominació comentades a l'apartat [6.1.2](#). Per realitzar aquestes proves primer s'han escrit tots els possibles casos i així assegurar-se que s'han tingut en compte totes les combinacions de nominacions. A continuació podem veure el codi font de cada una de les proves unitàries. Per l'estat inicial s'ha creat una instància d'una universitat en la qual el contracte bilateral és de 4 quadrimestres

```
class NominacioAnualTestCase(TestCase):
    contracte = []
    codi_universitat = 'CHNSHANGH02'
    universitat = []
    curs = []
    nominacions = []

    def setUp(self) -> None:
        self.universitat = Universitat(codi=self.codi_universitat, nom='Tongji University', acronim='TONGJI')
        self.contracte = ContracteBilateral(codi_universitat=self.codi_universitat, tipus='UPC-MON',
                                           any_acad=2021, quadris_in=4)
        self.nominacions.append(Application(name='Flor', last_name='Rimolo', institution=self.universitat,
                                           academic_course='2020/2021',
                                           period='A', email='flor@gmail.com', current_studies='GEI', token=None,
                                           double_degree='N', program='UPC-MON'
                                           ))

    def test_can_nominate(self):
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'A', -1), True)
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q1', -1), True)
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q2', -1), True)
```

Imatge 29: Prova unitària 1. Font: elaboració pròpia.

En aquesta primera prova, inicialment hi ha una nominació anual i es vol comprovar que es pot nominar tant un estudiant per un quadrimestre com un estudiant per tot l'any.

```
class NominacioQuadrimestralTestCase(TestCase):
    contracte = []
    codi_universitat = 'CHNSHANGH02'
    universitat = []
    curs = []
    nominacions = []

    def setUp(self) -> None:
        self.universitat = Universitat(codi=self.codi_universitat, nom='Tongji University', acronim='TONGJI')
        self.contracte = ContracteBilateral(codi_universitat=self.codi_universitat, tipus='UPC-MON',
                                           any_acad=2021, quadris_in=4)
        self.nominacions.append(Application(name='Flor', last_name='Rimolo', institution=self.universitat,
                                           academic_course='2020/2021',
                                           period='Q1', email='flor@gmail.com', current_studies='GEI', token=None,
                                           double_degree='N', program='UPC-MON'
                                           ))

    def test_can_nominate(self):
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'A', -1), True)
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q1', -1), True)
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q2', -1), True)
```

Imatge 30: Prova unitària 2. Font: elaboració pròpia.

A la segona prova, només es canvia la nominació inicial respecte la prova anterior. En aquest cas, és quadrimestral en comptes d'anyal, i es vol comprovar que es pot nominar tant un estudiant per un quadrimestre com un estudiant per tot l'any.

```
class DosNominacionsAnualsTestCase(TestCase):
    contracte = []
    codi_universitat = 'CHNSHANGH02'
    universitat = []
    curs = []
    nominacions = []

    def setUp(self) -> None:
        self.universitat = Universitat(codi=self.codi_universitat, nom='Tongji University', acronim='TONGJI')
        self.contracte = ContracteBilateral(codi_universitat=self.codi_universitat, tipus='UPC-MON',
                                           any_acad=2021, quadris_in=4)
        self.nominacions.append(Application(name='Flor', last_name='Rimolo', institution=self.universitat,
                                           academic_course='2020/2021',
                                           period='A', email='flor@gmail.com', current_studies='GEI', token=None,
                                           double_degree='N', program='UPC-MON'
                                           ))
        self.nominacions.append(Application(name='Flor', last_name='Rimolo', institution=self.universitat,
                                           academic_course='2020/2021',
                                           period='A', email='flor@gmail.com', current_studies='GEI', token=None,
                                           double_degree='N', program='UPC-MON'
                                           ))

    def test_cant_nominate(self):
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'A', -1), False)
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q1', -1), False)
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q2', -1), False)
```

Imatge 31: Prova unitària 3. Font: elaboració pròpia.

En aquesta tercera prova, inicialment tenim dues nominacions anuals. Es vol comprovar que, com que el contracte és de 4 quadrimestres, al tenir dues nominacions anuals ja no es poden nominar més estudiants.

```

class NominacioAnualIQudrimestralTestCase(TestCase):
    contracte = []
    codi_universitat = 'CHNSHANGH02'
    universitat = []
    curs = []
    nominacions = []

    def setUp(self) -> None:
        self.universitat = Universitat(codi=self.codi_universitat, nom='Tongji University', acronim='TONGJI')
        self.contracte = ContracteBilateral(codi_universitat=self.codi_universitat, tipus='UPC-MON',
                                           any_acad=2021, quadris_in=4)
        self.nominacions.append(Application(name='Flor', last_name='Rimolo', institution=self.universitat,
                                           academic_course='2020/2021',
                                           period='A', email='flor@gmail.com', current_studies='GEI', token=None,
                                           double_degree='N', program='UPC-MON'
                                           ))
        self.nominacions.append(Application(name='Flor', last_name='Rimolo', institution=self.universitat,
                                           academic_course=self.curs,
                                           period='Q2', email='flor@gmail.com', current_studies='GEI', token=None,
                                           double_degree='N', program='UPC-MON'
                                           ))

    def test_can_nominate(self):
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q1', -1), True)
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q2', -1), True)

    def test_cant_nominate(self):
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'A', -1), False)

```

Imatge 32: Prova unitària 4. Font: elaboració pròpia.

Per aquesta prova, inicialment hi ha una nominació anual i una altra quadrimestral. Es vol comprovar que només es pot realitzar una nominació quadrimestral més.

```

class NominacioAnualIDosQuadrimestralsTestCase(TestCase):
    contracte = []
    codi_universitat = 'CHNSHANGH02'
    universitat = []
    curs = []
    nominacions = []

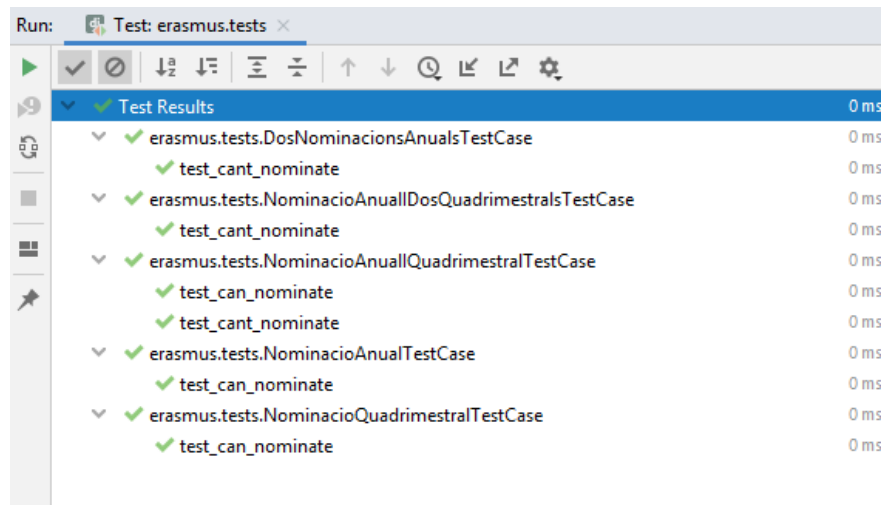
    def setUp(self) -> None:
        self.universitat = Universitat(codi=self.codi_universitat, nom='Tongji University', acronim='TONGJI')
        self.contracte = ContracteBilateral(codi_universitat=self.codi_universitat, tipus='UPC-MON',
                                           any_acad=2021, quadris_in=4)
        self.curs = '2020/2021'
        self.nominacions.append(Application(name='Flor', last_name='Rimolo', institution=self.universitat,
                                           academic_course=self.curs,
                                           period='A', email='flor@gmail.com', current_studies='GEI', token=None,
                                           double_degree='N', program='UPC-MON'
                                           ))
        self.nominacions.append(Application(name='Flor', last_name='Rimolo', institution=self.universitat,
                                           academic_course=self.curs,
                                           period='Q2', email='flor@gmail.com', current_studies='GEI', token=None,
                                           double_degree='N', program='UPC-MON'
                                           ))
        self.nominacions.append(Application(name='Flor', last_name='Rimolo', institution=self.universitat,
                                           academic_course=self.curs,
                                           period='Q1', email='flor@gmail.com', current_studies='GEI', token=None,
                                           double_degree='N', program='UPC-MON'
                                           ))

    def test_cant_nominate(self):
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'A', -1), False)
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q1', -1), False)
        self.assertEqual(can_nominate(self.nominacions, self.contracte, 'Q2', -1), False)

```

Imatge 33: Prova unitària 5. Font: elaboració pròpia.

A la cinquena i última prova, inicialment hi ha una nominació anual i dues quadrimestrals. Es vol comprovar que ja no es poden realitzar més nominacions.



Imatge 34: Resultat de les proves unitàries. Font: elaboració pròpia.

Com es pot veure, totes les proves han estat satisfactòries.

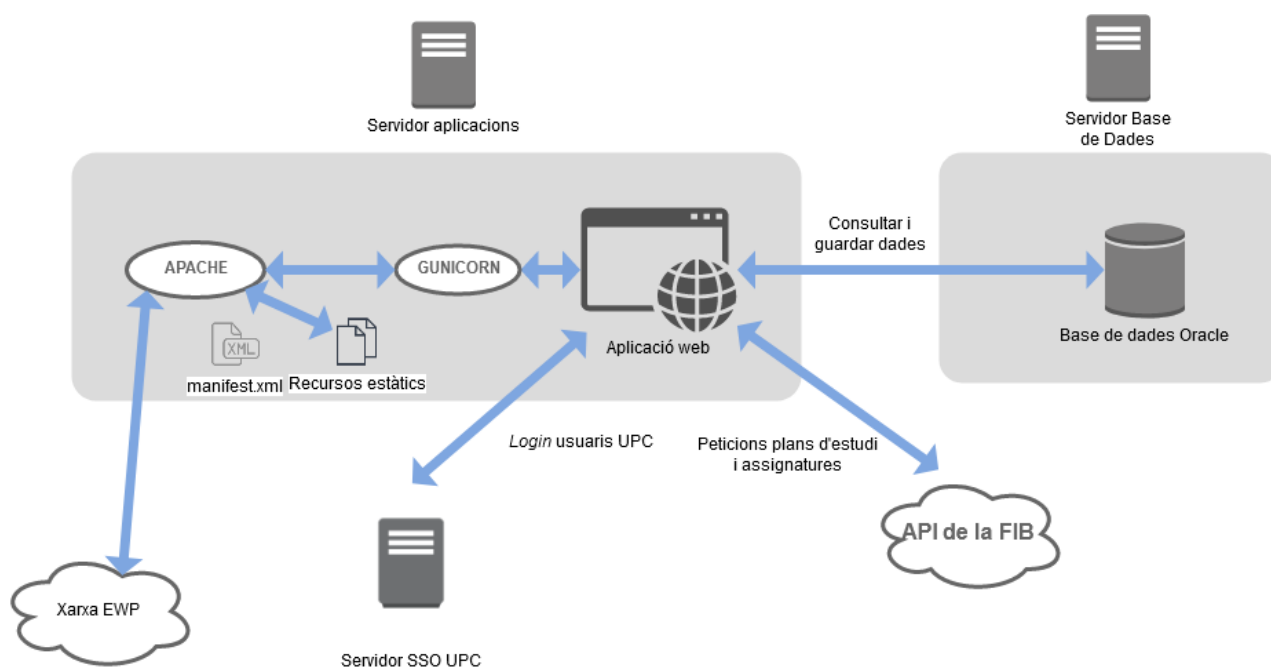
8. Posada en producció

Fins ara, el projecte s'ha provat i executat en un entorn local amb un servidor local que ens proporciona l'entorn de desenvolupament integrat que hem utilitzat, tal i com s'explica a l'apartat [7.1](#).

Així doncs, ens queda implantar el sistema al servidor on es farà públic pel seu ús. El servidor que s'ha escollit és una màquina nova, ja que la màquina on es troba l'aplicació antiga té una tecnologia obsoleta on no es poden instal·lar les versions més noves de les tecnologies que fem servir actualment, com per exemple la última versió de Python, que és la 3.7.

8.1. Arquitectura

A continuació podem veure un esquema que hem elaborat sobre l'arquitectura del nostre sistema a producció:



Imatge 35: Arquitectura del projecte a producció. Font: elaboració pròpia.

Tenim un servidor d'aplicacions en el qual es troba l'aplicació web configurada en un servidor HTTP WSGI per Python anomenat Gunicorn, que té suport natiu per les aplicacions desenvolupades amb

Django. Aquest servidor accedeix a l'arxiu executable de l'aplicació (*wsgi.py*), que conté la ruta a l'arxiu de configuració de l'aplicació (*Settings.py*).

A part del servidor Unicorn tenim un altre servidor HTTP, concretament Apache Server, que és el que reb totes les peticions de qualsevol de les aplicacions del node i posteriorment delega les peticions al servidor de l'aplicació corresponent que, en el nostre cas, seria el Unicorn. A més, aquest Apache és el que serveix el directori *static* de l'aplicació, ja que ho fa d'una manera més eficient que no pas el Unicorn, i l'arxiu *manifest.xml* que llegeix la xarxa EWP (a l'apartat [12.2](#) de l'annex s'explica en detall). També és el servidor que ens proporciona la connexió HTTPS.

Al servidor de base de dades tenim la base de dades Oracle, que hem hagut de reestructurar per adaptar-la a les necessitats de la nova aplicació. Ho explicarem al següent apartat.

8.2. Implantació

Un cop tenim definida l'arquitectura, només ens queda desplegar el projecte. Com s'ha comentat a l'apartat [7.1.7](#), s'utilitza Git com a eina de control de versions del projecte, per tant el primer pas és clonar el repositori on es troba. Seguidament hem d'instal·lar les llibreries externes que utilitza l'aplicació. Per fer-ho instal·larem *virtualenv*, que és un entorn virtual de Python que permet les instal·lacions de llibreries i paquets sense interferir amb els altres que pugui haver en el sistema, de manera que per exemple podem tenir la versió X d'una llibreria coexistint amb la versió Y de la mateixa, cada una en un entorn virtual diferent.

Abans d'executar l'aplicació, hem d'executar les migracions de la base de dades per crear o actualitzar aquelles taules gestionades pel propi Django. Com hem comentat a l'anterior apartat, hem hagut de realitzar una reestructuració de la base de dades que teníem anteriorment: concretament, hem creat un nou usuari per aïllar totes les taules de l'aplicació antiga i de l'aplicació d'escriptori i així no barrejar cap dada de manera que ens assegurarem utilitzar només aquelles taules que fa servir la nova aplicació. Com que hi ha taules que es compartiran amb l'aplicació d'escriptori, hem decidit crear un sinònim cap a aquestes, que estableix una dependència amb la taula a la qual creem l'àlies.

També hem hagut de migrar les dades de la taula antiga a la nova. Això s'ha fet per dues raons:

1. La taula antiga tenia una estructura que hem considerat que es podia optimitzar. Per posar un exemple, tenia un atribut que guardava el nom de la universitat origen a part de l'atribut amb el codi de la universitat, a partir del qual es pot establir una clau forana cap a la taules que guarda informació sobre les universitats tot estalviant-nos tenir una columna més.

2. La nova taula la gestionarà Django. En comptes de fer manualment els *inserts* en SQL dintre de l'aplicació, deixem que el propi *framework* actualitzi la taula de nominacions i sol·licituds, de manera que si volíem tenir a la mateixa taula les dades antigues, necessitàvem migrar-les.

Així doncs, hem implementat un *script* en Python que realitza la migració de les dades de la taula antiga a la nova, adaptant-les a l'estructura de l'aplicació.

Un cop tenim configurada la base de dades, ja podem executar l'aplicació.

9. Conclusions

Finalment hem aconseguit una aplicació web que s'adapta completament als requisits donats pel client i que compleix els objectius plantejats inicialment.

Amb el sistema que hem desenvolupat hem resolt els problemes descrits inicialment. Posem fi a la problemàtica de les nominacions per correu electrònic: ara tenim un sistema de nominacions ben definit amb el qual es pot saber en quin estat es troben en tot moment i que, a més, informa a la universitat origen quan ha excedit el nombre permès de nominacions. El mateix passa amb les sol·licituds d'estàncies acadèmiques: els estudiants podran saber en quin estat es troba la seva sol·licitud en tot moment i no ha d'esperar a que arribin correctament els documents requerits per correu postal, sinó que des de la pròpia aplicació els podrà pujar, la qual cosa provoca un estalvi de paper molt gran (recordem que cada curs venen uns 120 estudiants i que, per cada un d'ells, ens estalviem uns 10 fulls de paper). També s'han afegit noves petites funcionalitats que faciliten la feina a l'Àrea de Relacions Internacionals, com per exemple poder editar les assignatures que no estan obertes pels estudiants *incoming* i generar automàticament la carta d'acceptació.

A més, hem pogut implementar una prova de concepte de la xarxa EWP perquè, de cara al futur, serveixi com a referència per veure com funciona i com s'ha d'implementar. A l'apartat [12.2](#) de l'annex hem documentat aquesta prova de concepte.

Personalment, aquest projecte m'ha servit per adquirir més experiència en el disseny i desenvolupament d'aplicacions web i també per posar en pràctica els coneixements obtinguts durant tot el Grau. Al tractar-se d'un projecte real, he après a solventar alguns problemes que han sorgit com la desviació en el temps, a adaptar els requisits del client en tot moment durant el desenvolupament del projecte i a més he après els diferents passos per posar-lo en producció.

També he ampliat els meus coneixements sobre Python i Django, un dels *frameworks* més utilitzats en la creació d'aplicacions web.

10. Treball futur

Durant el desenvolupament del projecte s'ha parlat de diverses funcionalitats que, en un futur, es podrien incorporar a l'aplicació web, com per exemple:

- Implementar l'API Institutions [\[29\]](#) de la xarxa Erasmus Without Paper per obtenir informació sobre les universitats amb les quals tenim convenis, com l'adreça o les persones de contacte. Com s'ha comentat a l'apartat [2.3.4](#), actualment aquest projecte europeu no compta amb moltes universitats que l'implementin, per la qual cosa aquesta possible funcionalitat tindrà sentit quan es pugui extreure les dades de totes o la majoria de les universitats que tenen un acord bilateral amb la FIB.
- Utilitzar eduGAIN [\[30\]](#) com a sistema d'inici de sessió pels estudiants i les universitats *partner*. Aquest sistema és similar al de l'SSO UPC, però a nivell europeu, de manera que amb les credencials que utilitzen aquests usuaris per autenticar-se als sistemes de la seva institució també podrien entrar a la nostra aplicació.
- Poder editar les plantilles de correu des de la pròpia aplicació web, ja que amb la solució proposada s'ha d'accedir a la base de dades per poder fer-ho.

Aquestes idees són només algunes de les diverses que es poden incorporar. Volem que durant el futur desenvolupament del sistema aquest pugui millorar i incorporar noves funcionalitats que s'adaptin a les necessitats dels usuaris en cada moment, i així tenir un sistema que es vagi actualitzant en el temps.

11. Bibliografia

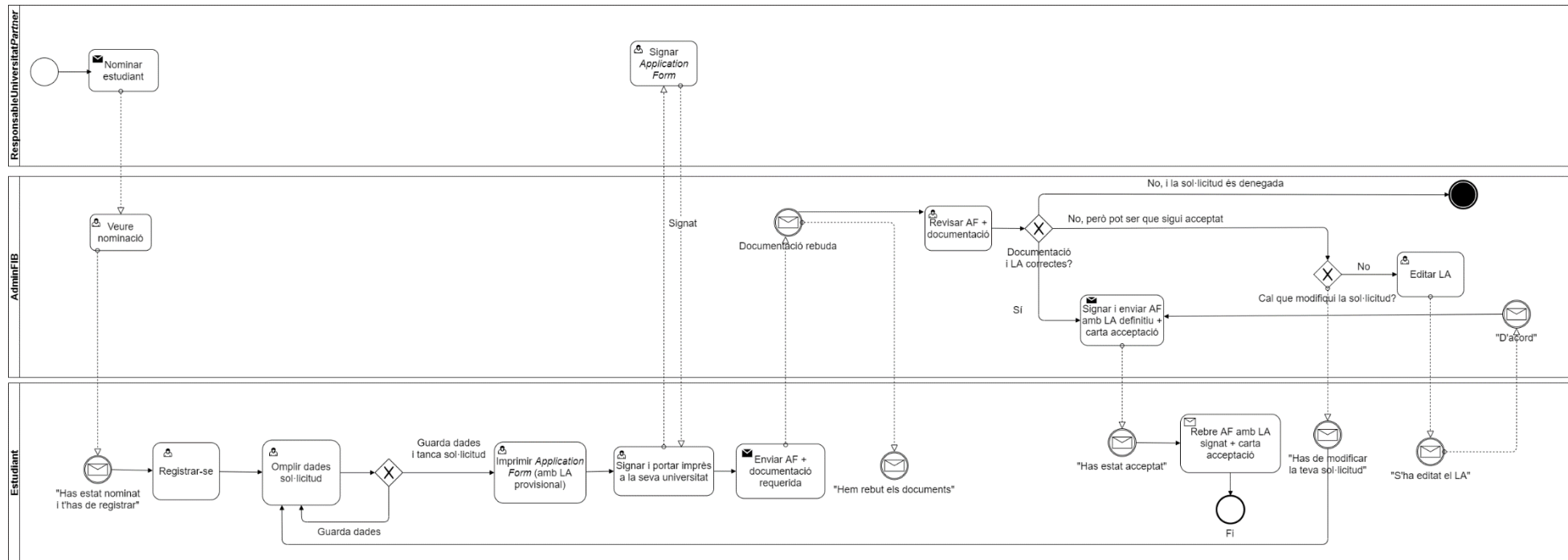
- [1] inLab FIB [en línia]. Disponible a: <https://inlab.fib.upc.edu/ca/> [Consultat el 13 d'octubre de 2019]
- [2] *Erasmus Without Paper* [en línia]. Disponible a: <https://www.erasmuswithoutpaper.eu/> [Consultat el 22 de febrer de 2019]
- [3] *Aliances Internacionals* [en línia]. Disponible a: <https://www.fib.upc.edu/ca/mobilitat/aliances-internacionals> [Consultat el 20 de febrer de 2019]
- [4] Solé Vives, Miquel (2009). *Eines de suport per a la gestió integrada de les RRll de la FIB* [en línia]. Disponible a: https://upcommons.upc.edu/bitstream/handle/2099.1/7765/PFC_Miquel_Sol%c3%a9.pdf?sequence=1&isAllowed=y [Consultat el 20 de febrer de 2019]
- [5] *Universidad Carlos III de Madrid. Nomination* [en línia]. Disponible a: <https://aplicaciones.uc3m.es/nominacionesIn/entrada.do> [Consultat el 20 de febrer de 2019]
- [6] European University Foundation (2017). *Erasmus Without Paper Final Conference* [en línia]. Disponible a: <https://www.youtube.com/watch?v=fjGAKQgFQRc> [Consultat el 22 de febrer de 2019]
- [7] European University Foundation (2018). *Erasmus Without Paper Network conference - EWP 2.0. (1/2)* [en línia]. Disponible a: <https://www.youtube.com/watch?v=SYGA1cmTg98> [Consultat el 22 de febrer de 2019]
- [8] *EWP Registry Service* [en línia]. Disponible a: <https://dev-registry.erasmuswithoutpaper.eu> [Consultat el 22 de febrer de 2019]
- [9] API de la FIB [en línia]. Disponible a: <https://api.fib.upc.edu> [Consultat el 9 de juliol de 2019]
- [10] SSO UPC [en línia]. Disponible a: <https://sso.upc.edu/> [Consultat el 14 d'octubre de 2019]
- [11] *Trello* [en línia]. Disponible a: <https://trello.com/> [Consultat el 25 de febrer de 2019]
- [12] Git [en línia]. Disponible a: <https://git-scm.com/> [Consultat el 24 de setembre de 2019]
- [13] Beedle, M. et al. (2001). *Principles behind the Agile Manifesto* [en línia]. Disponible a: <http://agilemanifesto.org/iso/en/principles.html> [Consultat el 25 de febrer de 2019]

- [14] Vicerectorat de Política Docent (2015). *Normativa de pràctiques externes de la Universitat Politècnica de Catalunya* [en línia]. Disponible a: <https://www.upc.edu/normatives/ca/butlleti-upc/hemeroteca/2014-2015/butlleti-upc-161/bupc-161-docs/docs-consell-govern/9.26-aprovacio-modificacio-normativa-practiques/@@download/file/9.26.%20Aprovacio%20modificacio%20normativa%20practiques.pdf> [Consultat el 8 de març de 2019]
- [15] PageGroup (2018). *Tendencias del mercado laboral. Tecnología* [en línia]. Disponible a: https://www.pagepersonnel.es/sites/pagepersonnel.es/files/PG_ER_IT_2018.pdf [Consultat el 8 de març de 2019]
- [16] Atlantic Systems Guid (2016). (Edition 18). *Volere Requirements Specification Template*.
- [17] Reglament General de Protecció de Dades [en línia] . Disponible a: https://apdcat.gencat.cat/web/.content/03-documentacio/Reglament_general_de_proteccio_de_dades/documents/2-RGPD-v7-Cat.pdf [Consultat el 4 d'octubre de 2019]
- [18] Django [en línia]. Disponible a: <https://www.djangoproject.com/> [Consultat el 23 de setembre de 2019]
- [19] Python [en línia]. Disponible a: <https://www.python.org/> [Consultat el 24 de setembre de 2019]
- [20] Bootstrap [en línia]. Disponible a: <https://getbootstrap.com/> [Consultat el 24 de setembre de 2019]
- [21] jQuery [en línia]. Disponible a: <https://jquery.com/> [Consultat el 24 de setembre de 2019]
- [22] Oracle Database [en línia]. Disponible a: <https://www.oracle.com/es/database/> [Consultat el 24 de setembre de 2019]
- [23] PyCharm [en línia]. Disponible a: <https://www.jetbrains.com/pycharm/> [Consultat el 24 de setembre de 2019]
- [24] DBeaver [en línia]. Disponible a: <https://dbeaver.io/> [Consultat el 24 de setembre de 2019]
- [25] GitLab UPC [en ínia]. Disponible a: <https://gitlab.upc.edu/> [Consultat el 24 de setembre de 2019]

- [26] GitLab [en línia]. Disponible a: <https://about.gitlab.com/> [Consultat el 24 de setembre de 2019]
- [27] Jinja2 [en línia]. Disponible a: <https://jinja.palletsprojects.com/en/2.10.x/> [Consultat el 2 d'octubre de 2019]
- [28] django_cas_ng [en línia]. Disponible a: <https://pypi.org/project/django-cas-ng/3.1.0/> [Consultat el 5 d'octubre de 2019]
- [29] Erasmus Without Paper (2017). *Institutions API* [en línia]. Disponible a: <https://github.com/erasmus-without-paper/ewp-specs-api-institutions/tree/v2.1.0> [Consultat el 6 d'octubre de 2019]
- [30] eduGAIN [en línia]. Disponible a: <https://edugain.org/> [Consultat el 13 d'octubre de 2019]

12. Annex

12.1. Diagrama del procés de l'aplicació actual



Imatge 36: Diagrama del procés en BPMN de l'aplicació actual. Font: elaboració pròpia.

12.2. Prova de concepte EWP

Durant la fase d'Autoaprenentatge, s'ha realitzat un estudi del funcionament de la xarxa EWP per elaborar una prova de concepte implementant l'API més senzilla de manera que, en un futur, serveixi com a referència per implementar funcionalitats d'aquesta xarxa.

Primer de tot, explicarem els conceptes bàsics per entendre exactament com funciona la xarxa EWP i què s'ha de fer per implementar-la.

12.2.1. Primers passos

Per utilitzar les APIs de la xarxa, necessitem un certificat SSL firmat per CA amb una clau pública RSA i una clau privada. Un cop el tenim, hem de crear un arxiu XML anomenat *manifest.xml*, en el qual declarem quines institucions (HEIs, de l'anglès *Higher Education Institutions*) allotja el nostre servidor, quines APIs implementem i quins certificats utilitzem per fer peticions als altres sistemes que implementen EWP. Aquest arxiu *manifest.xml* l'hem de publicar a un servidor, en nostre cas es troba a <https://mobility.fib.upc.edu/ewp/manifest.xml>.


```

▼<host>
  <ewp:admin-email>jaume.moral@fib.upc.edu</ewp:admin-email>
  <ewp:admin-email>florencia.rimolo@fib.upc.edu</ewp:admin-email>
▼<ewp:admin-notes>
  This is the Manifest File for Universitat Politècnica de Catalunya.
  </ewp:admin-notes>
▼<r:apis-implemented>
  ▼<discovery xmlns="https://github.com/erasmus-without-paper/ewp-specs-api-discovery/blob/stable-v5/manifest-entry.xsd" version="5.0.0">
    <url>https://mobility.fib.upc.edu/ewp/manifest.xml</url>
  </discovery>
  </r:apis-implemented>
▼<mf:institutions-covered xmlns="https://github.com/erasmus-without-paper/ewp-specs-api-registry/tree/stable-v1">
  ▼<hei id="fib.upc.edu">
    <other-id type="pic">999976202</other-id>
    <other-id type="erasmus">E BARCELO03</other-id>
    <other-id type="erasmus-charter">28604-EPP-1-2014-1-ES-EPPKA3-ECHE</other-id>
    <name xml:lang="en">Universitat Politècnica de Catalunya</name>
  </hei>
  </mf:institutions-covered>
▼<client-credentials-in-use>
  ▼<certificate>
    MIIHRDCCBiYgAwIBAgIQCEiiGpXl3gJY4B8n1V8WOzANBgkqhkiG9w0BAQ0FADBkMQswCQYDVQQGEwJOTDEWMBQGA1UECBMNTm9vcnQ9sbGFuZDESMBAGA1UEBxMjQW1zdG
    </certificate>
  ▼<rsa-public-key>
    MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAA4PrIyxp0VWYxXNKzeSUZWii09vwZPY9cTLMVQ68TS8Ygdx+V30JnR4dFzTD1AVMxqLLJj4FrXYSywg6sFoS6CPtL6d
    </rsa-public-key>
  </client-credentials-in-use>
▼<server-credentials-in-use>
  ▼<rsa-public-key>
    MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAA4PrIyxp0VWYxXNKzeSUZWii09vwZPY9cTLMVQ68TS8Ygdx+V30JnR4dFzTD1AVMxqLLJj4FrXYSywg6sFoS6CPtL6d
    </rsa-public-key>
  </server-credentials-in-use>
  ..

```

Imatge 37: Codi de l'arxiu manifest.xml. Font: elaboració pròpia.

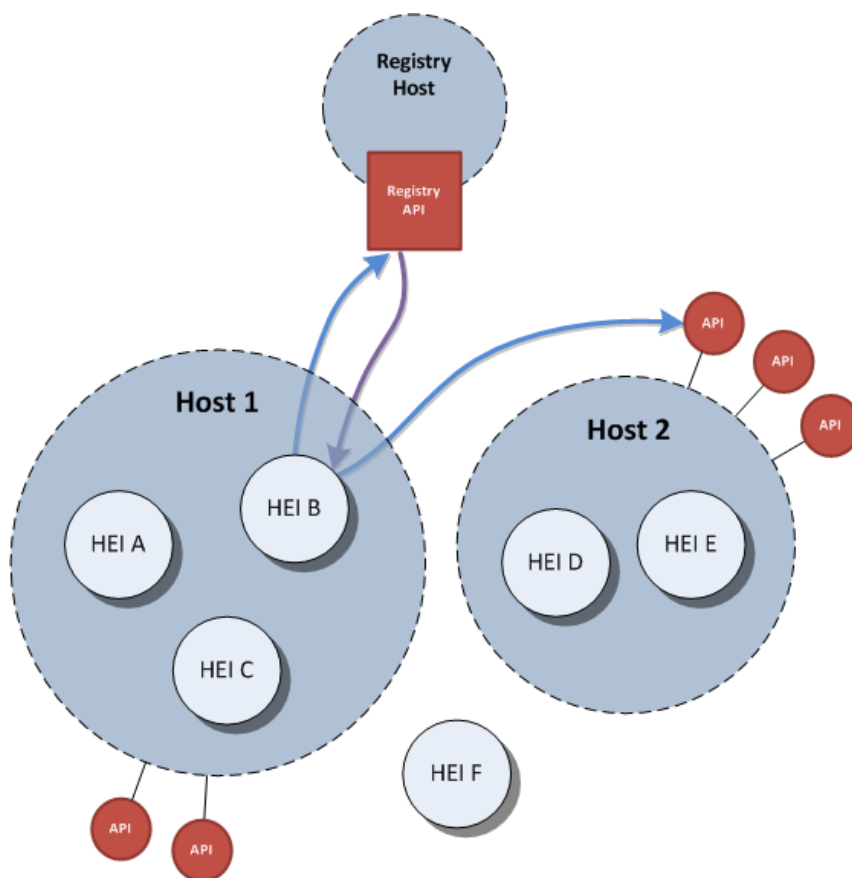
En aquesta imatge podem veure un fragment del nostre Manifest. A continuació expliquem què signifiquen algunes de les etiquetes XML que el formen:

- **<ewp:admin-email>**: Correu on s'envien notificacions sobre l'estat de la xarxa EWP.
- **<r:apis-implemented>**: Les URLs on es troben les diferents APIs que implementem. En el nostre cas tenim l'API Discovery, que és la primera API que han d'implementar totes les institucions per registrar-se (bàsicament serveix per indicar que estàs a la xarxa, ja que és la que conté l'arxiu *manifest.xml*).
- **<mf:institutions-covered>**: Aquí és on declarem les institucions que allotgem. Fixem-nos que en el camp *name* no hem posat el nom de la FIB sinó el de la UPC, ja que en el nostre cas l'identificador *E BARCELO03* correspon a tota la UPC i no a la nostra facultat. L'ideal seria tenir un ID per cada facultat, de manera que hi hagués un *host* per tota la UPC que allotgés a totes i així podríem afegir tants *hei_id* com facultats tingui la universitat.
- **<client-credentials-in-use>**: El certificat i la clau pública que utilitzem per fer peticions a les APIs de la xarxa.
- **<server-credentials-in-use>**: En algunes peticions que ens facin ens poden demanar el certificat del servidor.

Un cop tenim aquest arxiu amb la informació ben estructurada, s'ha d'enviar al correu que ens proporcionen la URL on es troba per ser registrats a la xarxa. Per saber si ens hem registrat correctament, busquem la nostra universitat al catàleg de les universitats que estan fent proves a EWP.

12.2.2. Arquitectura de la xarxa

En la següent imatge podem veure un esquema de l'estructura general dels diferents components de la xarxa EWP.



Imatge 38: Esquema de l'estructura general de la xarxa EWP. Font: <https://github.com/erasmus-without-paper/ewp-specs-architecture>

Observem que cada *host* implementa diverses APIs pels HEIs que allotja. Suposem que el HEI C que es troba al Host 1 vol fer una petició a una API obtenir informació proporcionada pel HEI E que es troba al Host 2. Inicialment, el HEI C no sap si el HEI E es troba a la xarxa EWP ni si implementa l'API a la qual vol fer la petició, de manera que necessita d'algun servei que li proporcioni aquestes dades. Es tracta del *Registry Service*, la part centralitzada de la xarxa, que permet accedir a la llista de *hosts* d'EWP amb les APIs que implementen ja que té accés a tots els Manifest.

Així doncs, el HEI C accedeix al *Registry Service* per obtenir la URL de l'API implementada del HEI E a la qual desitja cridar.

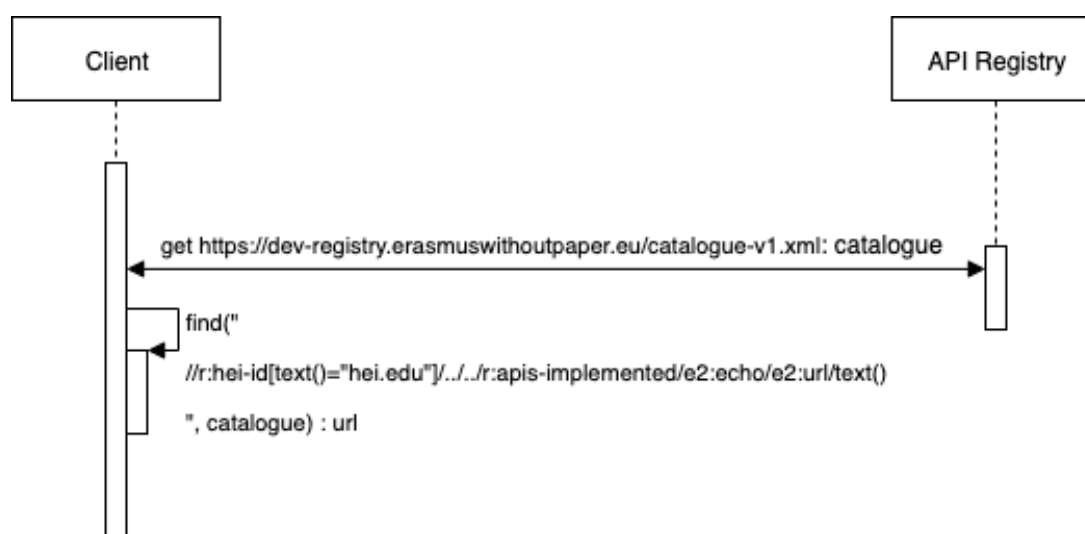
12.2.3. Com fer una petició

En aquest apartat explicarem què es necessita per realitzar una petició a una API de la xarxa EWP. Aquesta explicació serà en general, ja que cada API té uns requisits diferents, però en qualsevol cas el primer pas és consultar l'API Registry que implementa *Registry Service* per obtenir la URL de l'API que volem consultar. Per fer-ho, hem de realitzar una petició GET a la URL del *Registry*, que retorna una resposta en format XML que conté un catàleg de totes les institucions registrades a la xarxa amb les APIS que implementen. Un cop tenim la resposta, construïm una expressió XPath per filtrar l'XML per la institució que volem consultar.

Per exemple, si volem la URL de l'API Echo de la universitat que té com a HEI ID *hei.edu*, hauríem de filtrar per la següent expressió:

```
//r:hei-id[text()='hei.edu']/../r:apis-implemented/e2:echo/e2:url/text()
```

Hem elaborat el diagrama de seqüència d'aquesta petició al *Registry*:



Imatge 39: Diagrama de seqüència d'una petició a l'API Registry. Font: elaboració pròpia.

Amb això obtindríem la URL de l'API Echo, a la qual haurem de cridar amb els següents paràmetres:

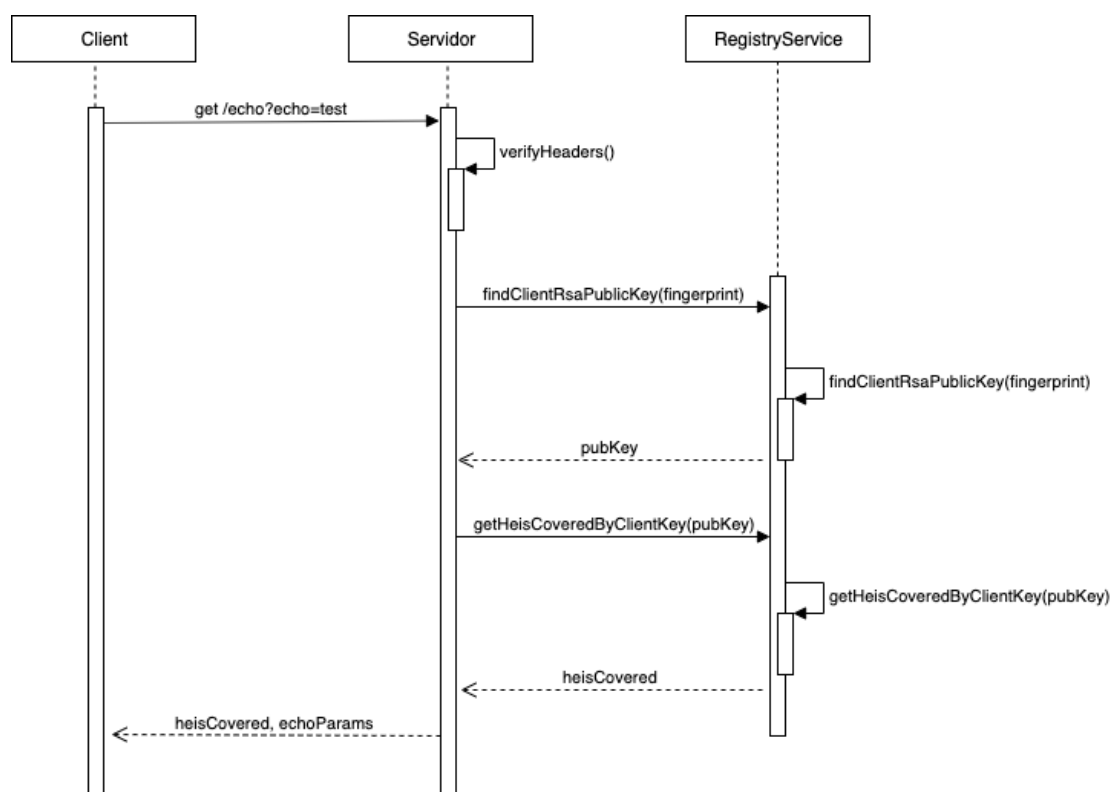
- La capçalera *Authorization* amb el mètode *Signature*. Això significa afegir els paràmetres:
 - *keyId*, amb el *fingerprint* SHA-256 codificat en hexadecimal de la clau pública RSA.
 - *algorithm*="rsa-sha256".
 - *headers*="(request-target) host date x-request-id digest".
- *Host*, amb el host al que fem la petició.

- *Date* o *Original-Date*.
- *X-Request-Id*, que conté un UUID (de l'anglès *Universally Unique Identifier*) en forma canònica.
- *Digest*, SHA-256 codificat en Base64 del cos de la petició.

Un cop tenim aquests paràmetres, ja podem cridar a la URL de l'API per obtenir les dades que desitgem, en format XML.

12.2.4. API Echo

La prova de concepte implementada s'ha realitzat fent una petició a dues APIs. La primera, l'API Echo de la Universitat de Varsòvia, ja que aquesta API serveix per entendre el funcionament d'una petició a la xarxa EWP. El que fa bàsicament és respondre amb els HEIs que allotja el *host* que fa la petició que, en el nostre cas, ens retornaria "fib.upc.edu". A més, permet passar-li paràmetres *query* anomenats *echo* amb un valor qualsevol, que ens els retornarà també a la resposta amb el valor que li hem proporcionat. En la imatge que es mostra a continuació podem veure el disseny d'una petició a l'API Echo.



Imatge 40: Diagrama del disseny d'una petició a l'API Echo. Font: elaboració pròpia.

Després d'afegir els paràmetres comentats a l'apartat anterior, fem la petició a la URL de l'API Echo de la universitat obtinguda de la petició a l'API Registry. El servidor que allotja aquesta institució, farà una comprovació d'aquests paràmetres verificant la signatura de la petició i demanarà al *Registry Service* que cerqui la clau pública del *host* que fa la petició per comprovar que estigui a la xarxa i així obtenir quins HEIs es troben allotjats en aquest. Un cop els obté, els retorna juntament amb els valors dels paràmetres *echo* de la URL de la petició inicial.

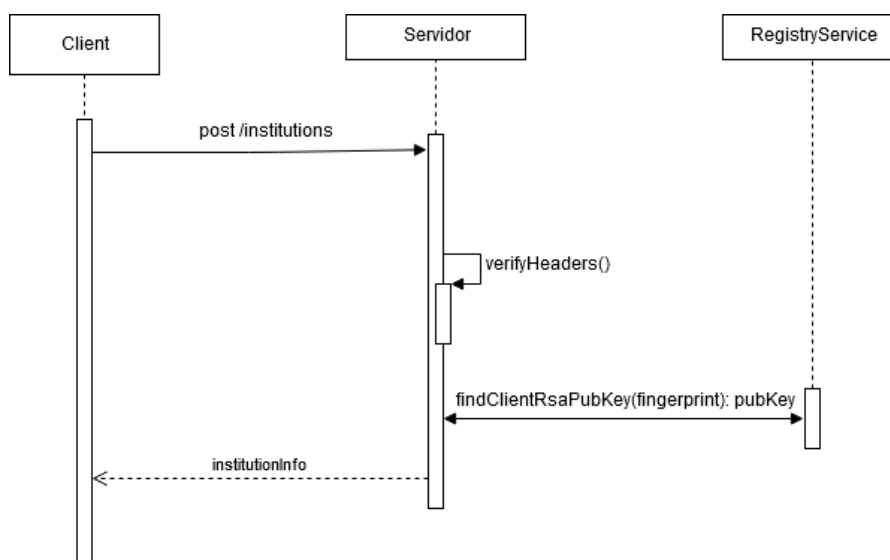
A la següent imatge podem veure la resposta a la petició a aquesta API:

```
<ns42:response xmlns="https://github.com/erasmus-without-paper/ewp-specs-api-registry/tree/stable-v1"
  <ns42:hei-id>fib.upc.edu</ns42:hei-id>
  <ns42:echo>hola</ns42:echo>
</ns42:response>
```

Imatge 41: Resposta de la petició a l'API Echo. Font: elaboració pròpia.

12.2.5. API Institutions

La segona API a la que hem fet una petició és l'API Institutions de la *Technische Universität Wien*, que retorna informació de la universitat registrada a la xarxa, com per exemple l'adreça, el logo de la institució, la URL per obtenir informació pels estudiants *incoming* d'aquesta institució o el personal de contacte. A la següent imatge veiem el diagrama de seqüència de la petició:



Imatge 42: Diagrama de seqüència d'una petició a l'API Institutions. Font: elaboració pròpia.

En aquest cas, fem una petició POST a la URL de l'API, obtinguda mitjançant la consulta a l'API Registry com s'ha explicat a l'apartat [12.2.3](#). En aquesta crida, hem d'afegir el cos de la petició en format *application/x-www-form-urlencoded*, amb un o més paràmetres que tenen com a clau *hei_id* i com a valor l'identificador del HEI dels quals volem obtenir informació.

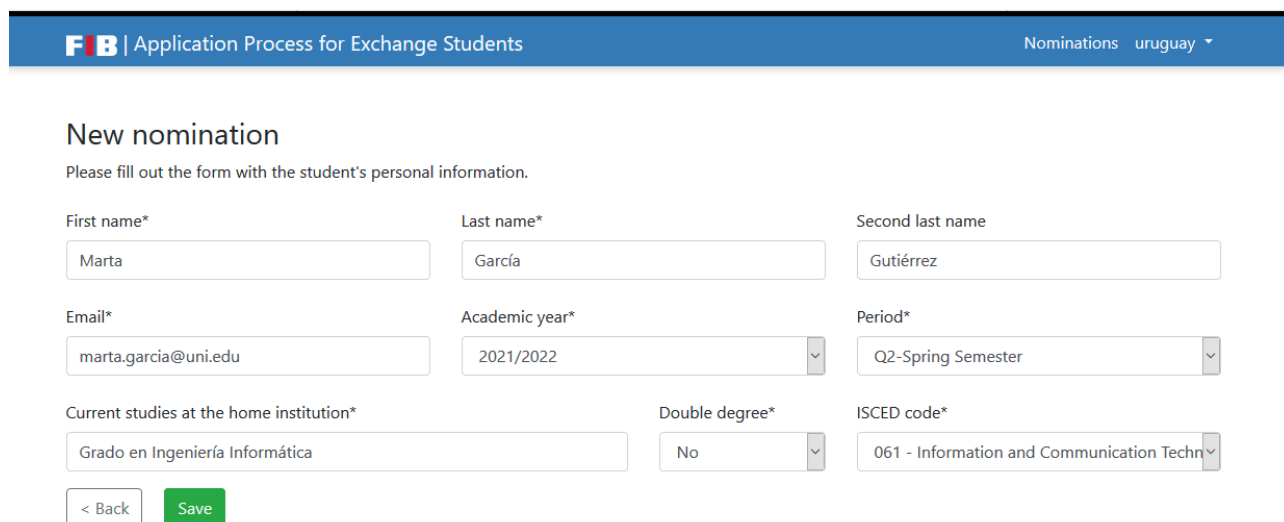
A continuació podem veure la resposta a la petició:

```
<institutions-response xmlns="https://github.com/erasmus-without-paper/ewp-specs-api-institutions/tree/stable-v2" xmlns:ns2="https://github.com/erasmus-wi
<hei>
  <hei-id>technikum-wien.at</hei-id>
  <other-id type="pic">998521105</other-id>
  <other-id type="erasmus">A WIEN20</other-id>
  <other-id type="euc">50550-IC-1-2007-1-AT-ERASMUS-EUCX-1</other-id>
  <name xml:lang="de">FACHHOCHSCHULE TECHNIKUM WIEN</name>
  <name xml:lang="en">FACHHOCHSCHULE TECHNIKUM WIEN</name>
  <abbreviation>WIEN20</abbreviation>
  <ns2:street-address>
    <ns2:recipientName>FACHHOCHSCHULE TECHNIKUM WIEN</ns2:recipientName>
    <ns2:addressLine>Höchstädtplatz 6</ns2:addressLine>
    <ns2:postalCode>1200</ns2:postalCode>
    <ns2:locality>Wien</ns2:locality>
    <ns2:region>Wien</ns2:region>
    <ns2:country>AT</ns2:country>
  </ns2:street-address>
  <ns2:mailing-address>
    <ns2:recipientName>FACHHOCHSCHULE TECHNIKUM WIEN</ns2:recipientName>
    <ns2:addressLine>Höchstädtplatz 6</ns2:addressLine>
    <ns2:postalCode>1200</ns2:postalCode>
    <ns2:locality>Wien</ns2:locality>
    <ns2:region>Wien</ns2:region>
    <ns2:country>AT</ns2:country>
  </ns2:mailing-address>
  <website-url>https://www.technikum-wien.at/en/</website-url>
  <logo-url>https://www.service4mobility.com/mobility_test/images/instImages/EAi7nVmhC61T5hWhVSJoh2LEDq7Wcy59Bg0Ly5gf7NP6.jpg</logo-url>
  <mobility-factsheet-url>https://www.service4mobility.com/mobility_test/GetStaticUploadFileServlet?match=c543add7e197c70175cc3d1ed2efda67df0c65c32c063e
  <ns3:contact>
    <ns3:contact-name>Kriz Agnes</ns3:contact-name>
    <ns3:person-given-names>Agnes</ns3:person-given-names>
    <ns3:person-family-name>Kriz</ns3:person-family-name>
    <ns3:person-gender>2</ns3:person-gender>
  </ns3:contact>
  <ns4:phone-number>
```

Imatge 43: Resposta de la petició a l'API Institutions. Font: elaboració pròpia.

12.3. Captures de pantalla de l'aplicació

En aquest apartat adjuntem diverses captures d'algunes de les pantalles de l'aplicació.



FIB | Application Process for Exchange Students Nominations uruguay ▾

New nomination

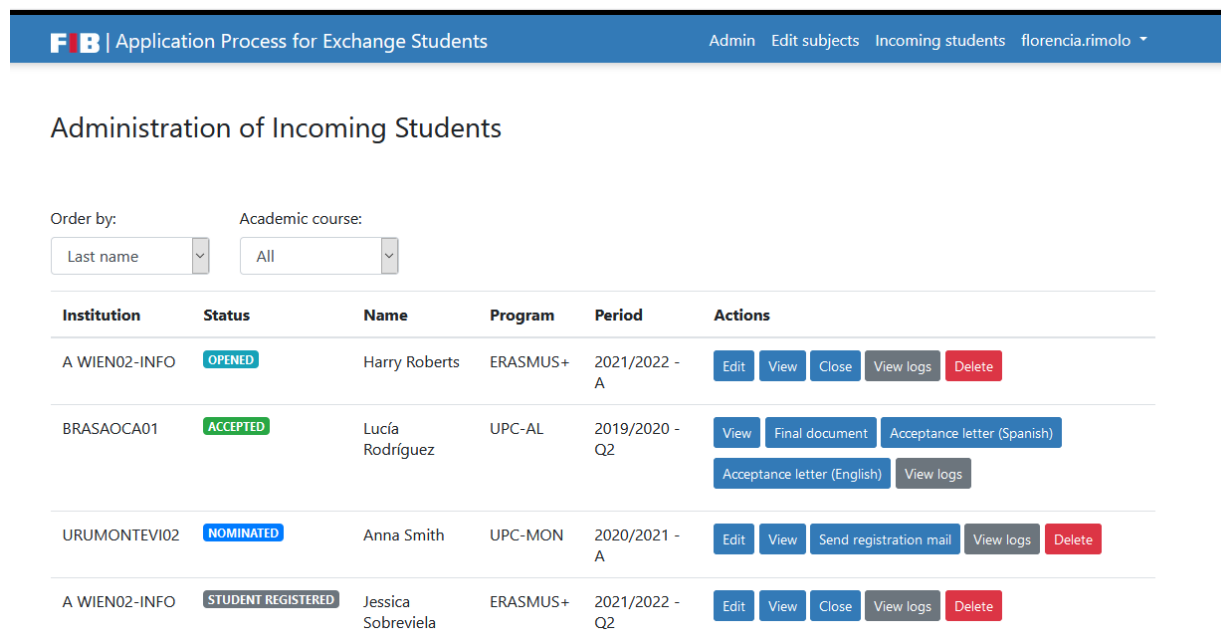
Please fill out the form with the student's personal information.

First name* Last name* Second last name

Email* Academic year* Period*

Current studies at the home institution* Double degree* ISCED code*

Imatge 44: Pantalla formulari de nominació. Font: elaboració pròpia.



FIB | Application Process for Exchange Students Admin Edit subjects Incoming students florencia.rimolo ▾

Administration of Incoming Students

Order by: Academic course:




Institution	Status	Name	Program	Period	Actions
A WIEN02-INFO	OPENED	Harry Roberts	ERASMUS+	2021/2022 - A	<input type="button" value="Edit"/> <input type="button" value="View"/> <input type="button" value="Close"/> <input type="button" value="View logs"/> <input type="button" value="Delete"/>
BRASAOCA01	ACCEPTED	Lucía Rodríguez	UPC-AL	2019/2020 - Q2	<input type="button" value="View"/> <input type="button" value="Final document"/> <input type="button" value="Acceptance letter (Spanish)"/> <input type="button" value="Acceptance letter (English)"/> <input type="button" value="View logs"/>
URUMONTEVI02	NOMINATED	Anna Smith	UPC-MON	2020/2021 - A	<input type="button" value="Edit"/> <input type="button" value="View"/> <input type="button" value="Send registration mail"/> <input type="button" value="View logs"/> <input type="button" value="Delete"/>
A WIEN02-INFO	STUDENT REGISTERED	Jessica Sobreviela	ERASMUS+	2021/2022 - Q2	<input type="button" value="Edit"/> <input type="button" value="View"/> <input type="button" value="Close"/> <input type="button" value="View logs"/> <input type="button" value="Delete"/>

Imatge 45: Pantalla de nominacions i sol·licituds. Font: elaboració pròpia.

Edit Subjects

Here you can edit the subjects that are not offered to the incoming students:

Choose a subject

CODE	NAME	ECTS CREDITS	REMOVE
ASO	Operating Systems Administration	6.0	
PTI	Information Technology Project	6.0	
TEB-MEI	Techniques and Tools for Bioinformatics	3.0	

Save

Imatge 46: Pantalla per editar les assignatures no ofertes. Font: elaboració pròpia.

FIB | Application Process for Exchange Students

My application harry.roberts ▾

Here you can see a summary of your application/s, to view the full data click the 'View' button. The data shown below can not be modified, but you can click on 'Edit' to modify the other information. This information data we collect in order to process your registration will be used only by ourselves at University environment. If you do not end the application, your data will be deleted from the system within a maximum period of 6 months. Please, if there is any error you can report it in rel.int@fib.upc.edu.

My application

Institution: Technische Universitat Wien - Faculty of Informatics

Name: Harry

Last name: Roberts

Second last name:

Email: harry.roberts@example.edu

Current studies: Bachelor Degree in Computer Science

Period: 2020/2021 - Annual

Status: OPENED

Edit

View

Imatge 47: Pantalla amb resum d'una sol·licitud. Font: elaboració pròpia.

Application Form of Harry Roberts

Home institution

Country*

Austria

Institution*

Technische Universität Wien - Faculty of Informatics

School/department*

FIB

Student exchange coordinator

First name*

Lukas

Last name*

Müller

Phone number (with prefix)*

+43 123456789

Fax (with prefix)*

+43 123456789

Email*

lukas.muller@example.edu

This address will be where we send the official documents. In case the contact person was different, add it or add other information you think necessary at the end (extra info).

Street and number*

Imatge 48: Pantalla del formulari de sol·licitud (1). Font: elaboració pròpia.

Street and number*

Street, 90

City and country*

Vienna (Austria)

ZIP/Postal code*

1040

Extra information

Student's personal information

First name*

Harry

Last name*

Roberts

Second last name

Gender*

Male

Birth date*

1994-03-16

Birth place*

Vienna

Nationality*

Austrian

ID Card/Passport number*

P1234567

Email*

harry.roberts@example.edu

Contact phone number*

+43 01234567

Imatge 49: Pantalla del formulari de sol·licitud (2). Font: elaboració pròpia.

FIB | Application Process for Exchange Students My application harry.roberts ▾

Permanent address

Street and number*

City*

Country*

ZIP/Postal code*

Phone number (with prefix)*

Language competence

Mother tongue*

Language of instruction at home institution*

Language certificate in PDF format*
[language_certificate.pdf](#) Remove

LANGUAGE	SPEAKING	READING	WRITING	LISTENING
ENGLISH	<input type="text" value="Excellent"/>	<input type="text" value="Excellent"/>	<input type="text" value="Excellent"/>	<input type="text" value="Excellent"/>
CATALAN	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>

Imatge 50: Pantalla del formulari de sol·licitud (3). Font: elaboració pròpia.

FIB | Application Process for Exchange Students My application harry.roberts ▾

SPANISH

CATALAN

Academic information and proposed Learning Agreement

Please first of all introduce the URL where we can find the syllabus of every one of the subjects you have already coursed at your home institution and which must be in your transcript of records attached. The information found should be in English (or Spanish, if possible). You can add different URLs in case your subjects syllabus are listed in different locations.

URL*
 Add

Period*

Academic year*

Comments

Imatge 51: Pantalla del formulari de sol·licitud (4). Font: elaboració pròpia.

FIB | Application Process for Exchange Students
My application harry.roberts ▾

The list of subjects available is filtered by the studies you have selected and the semester or year when you're coming. Subjects requirements are also considered, so you will not be able to choose a subject if it is the prerequisite of another. Remember the learning agreement has to be a maximum of 30 (exceptionally 36) ECTS credits per semester. On the following link you can find all the information about the subjects, please read it carefully before filling the table below: <https://www.fib.upc.edu/en/mobility/incoming/how-apply>

Subjects*

CODE	NAME	LANGUAGE	SEMESTER	ECTS CREDITS	REMOVE
ER	Requirements Engineering	Catalan, Spanish	Q2 ▾	6.0	-
PAR	Parallelism	Catalan, Spanish, English	Q1 ▾	6.0	-

Choose a subject
 ▾

Currently enrolled subjects (at home institution) in PDF format*

[enrolled_subjects.pdf](#) Remove

Curriculum in PDF format*

[curriculum.pdf](#) Remove

Current academic record in PDF format. In case you are a Master student, please upload the academic record of the bachelor degree (or similar e.g. the Diploma Supplement) too*

Examinar... No se ha seleccionado ningún archivo.

[academic_record_bachelor.pdf](#) Remove

Imatge 52: Pantalla del formulari de sol·licitud (5). Font: elaboració pròpia.

FIB | Application Process for Exchange Students
My application harry.roberts ▾

Currently enrolled subjects (at home institution) in PDF format*

[enrolled_subjects.pdf](#) Remove

Curriculum in PDF format*

[curriculum.pdf](#) Remove

Current academic record in PDF format. In case you are a Master student, please upload the academic record of the bachelor degree (or similar e.g. the Diploma Supplement) too*

Examinar... No se ha seleccionado ningún archivo.

[academic_record_bachelor.pdf](#) Remove

In case you're planning to do your final project at FIB, please choose the number of credits of the project.

Final project*

☐ No project
☒ 18 ECTS Bachelor project
☐ 18 ECTS Master project
☐ 30 ECTS Master project

In case already contacted, UPC/FIB professor

< Back
Save
Save and close

Imatge 53: Pantalla del formulari de sol·licitud (6). Font: elaboració pròpia.

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona**LEARNING AGREEMENT****Student's name:** Harry Roberts**Academic year:** 2020/2021**Period:** Annual**Degree:** Bachelor Degree in Informatics Engineering

Code	Subject name	Semester	ECTS credits
ER	Requirements Engineering	Q2	6.0
PAR	Parallelism	Q1	6.0



M. Teresa Abad Soriano
Vice-Dean for Students
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya



12 October 2019

Letter of Acceptance

Dear Harry Roberts ,

I am pleased to inform that you have been accepted at **Facultat d'Informàtica de Barcelona** of the **Universitat Politècnica de Catalunya** as an exchange student for the Annual period of the academic year 2020/2021.

The academic period starts the 4 September 2020 and ends the 20 June 2021, including exams, but you are recommended to be here for the Orientation week on September 2.

You have at your disposal the service of Resa Housing or Housinganywhere, which provides and manages accommodation for foreign students.

Best regards,



M. Teresa Abad Soriano
Vice-Dean for Students
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Note: The University General Code of the Spanish Ministry of Education is 024.

Imatge 55: Carta d'acceptació generada per l'aplicació. Font: elaboració pròpia.